

Bases para Construir Modelos de Conhecimento Coletivo no Paradigma Simbólico

DISSERTAÇÃO DE MESTRADO

Nélio Duarte Pereira Ramos

MESTRADO EM ENGENHARIA INFORMÁTICA



UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

julho | 2016

**Bases para Construir Modelos
de Conhecimento Coletivo
no Paradigma Simbólico**

DISSERTAÇÃO DE MESTRADO

Nélio Duarte Pereira Ramos
MESTRADO EM ENGENHARIA INFORMÁTICA

ORIENTADOR
Eduardo Leopoldo Fermé

Agradecimentos

Aos meus pais que sempre me inculcaram o gosto pelo saber.

Ao meu orientador, Professor Eduardo Fermé, pela sua disponibilidade e, acima de tudo, paciência que teve para comigo.

Palavras-Chave

Agentes Inteligentes
Argumentação e Revisão de Crenças
Inteligência Artificial
Representação do Conhecimento
Sistemas Multiagentes
Teoria da Decisão

Keywords

Intelligent Agents
Argumentation and Belief Revision
Artificial Intelligence
Knowledge Representation
Multiagent Systems
Decision Theory

BASES PARA CONSTRUIR MODELOS DE
CONHECIMENTO COLETIVO NO
PARADIGMA SIMBÓLICO

JÚRI

PROFESSOR DOUTOR EDUARDO LEOPOLDO FERMÉ

PROFESSORA DOUTORA KAROLINA BARAS

PROFESSOR DOUTOR SERGI BERMÚDEZ I BADIA

Resumo

Na idade da informação a capacidade de recolher, escolher e armazenar a informação relevante é um propósito tão nobre como complexo. A computação ubíqua permite que o conhecimento esteja distribuído por diversos locais e intervenientes. Na Inteligência Artificial o conceito de Agente Inteligente faz já parte do léxico comum e a sua utilidade tem vindo a ser demonstrada para resolução de diversos problemas. Para tal contribuiu a sua capacidade de sistematizar o processo de trabalhar a informação. A interligação entre estes agentes cria um *conhecimento comum* a todo o grupo.

Neste trabalho são propostas bases que permitem a construção de um Modelo de conhecimento coletivo. Estas bases permitem uma visão para a representação do conhecimento num grupo de agentes com recurso a ferramentas do paradigma simbólico e através da revisão de crenças e argumentação para a dinâmica de conhecimento entre os elementos de um grupo.

Abstract

In the information age the ability to collect, select and store relevant information is a purpose as noble as complex. Ubiquitous computing enables knowledge distributed on various locations and actors. In Artificial Intelligence the concept of Intelligent Agent is already part of the common lexicon and its usefulness has been demonstrated to solve various problems. Contributing to this is the ability to systematize the process of information manipulating. The interconnection between these agents creates a *common knowledge* of the whole group.

This work proposal a basis to allow the construction of a collective knowledge model. These allows for a vision of knowledge representation in an agent group using the symbolic paradigm tools and through the belief revision and argumentation for knowledge dynamic between the elements of the group.

Índice

1.	INTRODUÇÃO	1
1.1.	ORGANIZAÇÃO	2
2.	AGENTES INTELIGENTES.....	4
2.1.	INTELIGÊNCIA ARTIFICIAL.....	4
2.2.	AGENTES	7
2.2.1.	PROPRIEDADES	9
2.2.2.	ARQUITETURAS	9
2.3.	CONCLUSÃO.....	11
3.	SISTEMAS MULTIAGENTES	12
3.1.	COMUNICAÇÃO	13
3.2.	COOPERAÇÃO/COMPETIÇÃO.....	14
3.3.	PERSPETIVAS	15
3.4.	CONCLUSÃO.....	15
4.	PARADIGMAS INTELIGÊNCIA ARTIFICIAL.....	16
4.1.	PARADIGMA CONEXIONISTA	18
4.2.	PARADIGMA EVOLUCIONISTA	20
4.3.	PARADIGMA SIMBÓLICO.....	22
4.4.	CONCLUSÃO.....	22
5.	INTELIGÊNCIA DE GRUPO.....	24
5.1.	BIOCOMPUTAÇÃO	24
5.2.	INSETOS SOCIAIS	25
5.3.	ANT COLONY OPTIMIZATION	25
5.4.	BEE ALGORITHM.....	27
5.5.	PSO.....	28
5.6.	MODELO SECI.....	29
5.7.	CONCLUSÃO.....	31

6. CONHECIMENTO COLETIVO	32
6.1. REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO	32
6.2. ABORDAGEM CONEXIONISTA.....	33
6.3. ABORDAGEM EVOLUCIONÁRIA.....	33
6.4. ABORDAGEM SIMBÓLICA	34
6.5. CONCLUSÃO.....	35
7. REPRESENTAÇÃO DO CONHECIMENTO	38
7.1. CONHECIMENTO: REPRESENTAÇÃO E RACIOCÍNIO	38
7.1.1. CONHECIMENTO.....	38
7.1.2. REPRESENTAÇÃO	39
7.1.3. RACIOCÍNIO.....	39
7.1.4. REPRESENTAÇÃO E RACIOCÍNIO.....	40
7.1.5. O PAPEL DA LÓGICA	42
7.2. LINGUAGEM DA LÓGICA DE PRIMEIRA ORDEM	43
7.2.1. LINGUAGEM.....	43
7.3. EXPRESSÃO DO CONHECIMENTO	49
7.4. REPRESENTAÇÃO ORIENTADA A OBJETOS.....	51
7.5. DESCRIÇÕES ESTRUTURADAS.....	54
7.5.1. LINGUAGEM DE DESCRIÇÃO	55
7.5.2. SIGNIFICADO E IMPLICAÇÃO.....	56
7.5.3. TAXONOMIA E CLASSIFICAÇÃO	59
7.6. HERANÇA	61
7.7. INDEFINIÇÃO, INCERTEZA E GRAUS DE CRENÇA	65
7.7.1. RACIOCÍNIO NÃO BOOLEANO	65
7.7.2. PROBABILIDADE OBJETIVA.....	66
7.7.3. PROBABILIDADE SUBJETIVA	67
7.7.4. INCERTEZA	70
7.8. CONCLUSÃO.....	71
8. DINÂMICA DO CONHECIMENTO	72
8.1. REVISÃO DE CRENÇAS	72
8.1.1. AGM	72
8.1.2. BASE DE CRENÇAS	75
8.2. ARGUMENTAÇÃO.....	77
8.2.1. PLATAFORMA ABSTRATA DE DUNG.....	78
8.2.2. DELP	78
8.3. REVISÃO DE CRENÇAS E ARGUMENTAÇÃO.....	78

8.3.1.	SISTEMAS DE MANUTENÇÃO DE VERDADE	79
8.3.2.	REVISÃO DE CRENÇAS E EPISTEMOLOGIA	79
8.3.3.	REVISÃO DE CRENÇAS E EXPLANAÇÕES DEDUTIVAS	79
8.3.4.	REVISÃO DE CRENÇAS ORIENTADA A DADOS	80
8.3.5.	REVISÃO PRIORITÁRIA POR ARGUMENTOS	80
8.3.6.	RECUPERAÇÃO E REINSTALAÇÃO RELACIONADA.....	80
8.4.	CONCETUALIZAÇÃO DO MODELO	81
8.4.1.	COMPARAÇÃO	82
8.4.2.	ARGUMENTAÇÃO NA REVISÃO DE CRENÇAS.....	82
8.4.3.	REVISÃO DE CRENÇAS NA ARGUMENTAÇÃO.....	83
8.5.	MODELO BDI	84
8.6.	CONCLUSÃO.....	86
9.	TEORIA DA DECISÃO.....	87
9.1.	ESCOLHA SOCIAL.....	89
9.2.	ESCOLHA SOCIAL COMPUTACIONAL	90
9.3.	CONCLUSÃO.....	92
10.	BASES PARA CONSTRUIR MODELOS DE CONHECIMENTO COLETIVO NO PARADIGMA SIMBÓLICO.....	93
10.1.	AGENTES	94
10.2.	REPRESENTAÇÃO E RACIOCÍNIO	95
10.3.	DINÂMICA DO CONHECIMENTO	95
10.4.	MODELO	96
10.5.	CONCLUSÃO.....	97
11.	CONCLUSÕES E TRABALHO FUTURO	98
12.	BIBLIOGRAFIA E REFERÊNCIAS	99

Índice Figuras

Figura 2.1 – Inteligência Artificial, campos antecedentes e aplicações	5
Figura 2.2 – Agente Genérico	8
Figura 2.3 – Arquitetura Deliberativa	10
Figura 2.4 – Arquitetura Reativa.....	10
Figura 2.5 – Arquitetura Híbrida	11
Figura 4.1 – Componentes de um Sistema de Inteligência Artificial por Sage.....	16
Figura 4.2 – Modelo simples de aprendizagem [19]	17
Figura 4.3 – Diagrama do Sistema Nervoso	18
Figura 4.4 – Neurónio [22].....	19
Figura 4.5 – Ativação de neurónio artificial segundo McCulloch e Pitts.....	19
Figura 4.6 – Cromossoma e gene [24]	21
Figura 5.1 – Percentagem da passagem das formigas num dos percursos, de [34]	26
Figura 5.2 – Passagem das formigas num dos percursos [35]	26
Figura 5.3 – Bee Algorithm	28
Figura 5.4 – Algoritmo PSO.....	29
Figura 5.5 – Modelo SECI [42]	30
Figura 6.1 – Troca de informação genética por crossover e mutação	34
Figura 7.1 – Mundo de 3 blocos	48
Figura 7.2 – Frame individual.....	52
Figura 7.3 – Frame genérica	52
Figura 7.4 – Rede de herança simples [45].....	62
Figura 7.5 – Herança derrotável [45]	62
Figura 7.6 – Curva de grau de um predicado vago [45]	70
Figura 8.1 – Estrutura BDI genérica	86
Figura 10.1 – Modelo de agente simplificado	94
Figura 10.2 – Modelo base	96

Índice Tabelas

Tabela 2.1 – Inteligência Artificial e campos relacionados.....	4
Tabela 2.2 – Definições de Agentes.....	8
Tabela 4.1 – Definições de Inteligência Artificial	16

Acrónimos

ENIAC – *Electronic Numerical Integrator and Computer*
EDVAC – *Electronic Discrete Variable Automatic Computer*
LISP – *List Processing*
XCON – *Expert Configurer*
NCSA – *National Center for Supercomputing Applications*
AIBO – *Artificial Intelligence Robot*
DARPA – *Defense Advanced Research Projects Agency*
ASIMO – *Advanced Step in Innovative Mobility*
IA/AI – *Inteligência Artificial / Artificial Intelligence*
SMA/MAS – *Sistema Multiagente / Multi-agent System*
SFS – *Sistema Físico de Símbolos*
SOA – *Swarm-based Optimisation Algorithms*
ACO – *Ant Colony Optimization*
PSO – *Particle Swarm Optimization*
SECI – *Socialization Externalization Internalization Combination*
KB – *Knowledge Base*
FOL – *First Order Logic*
DL – *Description Language*
DAG – *Directed Acyclic Graph*
DELP – *Defeasible Logic Programming*
TMS – *Truth Maintenance System*
ATMS – *Assumption-Based Truth Maintenance System*
WPA – *Warrant-prioritized argument revision*
DBR – *Data-oriented Belief Revision*
BDI – *Believe Desire Intention*

1. INTRODUÇÃO

A representação do conhecimento apresenta-se de grande importância pois desse há muito o armazenamento e passagem do conhecimento é um tema central na cultura dos povos. Desde os tempos em que se passava as histórias à volta da fogueira até à ubiquidade da informação presente nos dias de hoje. Uma característica deste processo será a distinção entre *crença* e *conhecimento*. Uma crença é algo que alguém acredita que poderá não ser ou não justificado. No mundo natural algo deste tipo pode ser exequível, mas em termos computacionais pode levar a consequências inesperadas de um estado contraditório no sistema. A representação do conhecimento e raciocínio sobre o mesmo tem sido um campo fértil na Inteligência Artificial. Pois envolve várias tecnologias tais como, por exemplo, inferência, planificação e aprendizagem que têm tido sucesso em várias aplicações. O raciocínio necessita que quem o efetua detenha uma representação do ambiente onde se insere. Dessa forma é necessário definir uma linguagem que seja capaz de representar este conhecimento. O paradigma simbólico parece fornecer a estrutura capaz de suportar as necessidades de tal questão. Os agentes conseguem deduzir várias informações a partir da inferência lógica. Estas informações podem ser retiradas do ambiente bem como da interação com outros agentes. Assim através da *Revisão de Crenças* os agentes podem incorporar estas novas informações no seu conhecimento. A *Argumentação* fornece a capacidade ao agente para tentar que os outros acomodem a sua visão.

Este trabalho parte do conceito de *Conhecimento Coletivo no Paradigma Simbólico*. Sendo assim descrevemos os pontos essenciais do conhecimento neste paradigma e o conhecimento coletivo. Apresentamos uma proposta de bases que sustentem uma plataforma para a representação do conhecimento coletivo com recurso a ferramentas do paradigma simbólico. O Paradigma Simbólico integra a Inteligência como computação; um conjunto de processos que atuam sobre estruturas simbólicas. A grande vantagem deste é que suporta as três características do Conhecimento: ser verdadeiro; ser acreditável e ser demonstrável. O conhecimento coletivo emerge da interação dos elementos de um grupo social. Este pode ser observado no comportamento dos insetos sociais tais como formigas e abelhas.

A pouca menção desta visão advém da utilização do Paradigma Simbólico nos sistemas Especialistas em detrimento do processamento de problemas genéricos, nos quais apresenta algumas dificuldades. Além disso a Inteligência Artificial tem-se centrado, nos últimos anos na aprendizagem máquina e em algoritmos estatísticos. Tendo o processamento de informação através de manipulação de símbolos no nível semântico e a construção de modelos de estruturas cognitivas ficado um pouco para trás no tempo. Para responder a tal é necessário a representação do conhecimento, raciocínio e dinâmica do conhecimento. Estes três pontos serão as bases para sustentar o modelo proposto neste trabalho.

1.1. ORGANIZAÇÃO

Este trabalho está organizado em três partes. A primeira, que se compõe dos capítulos 2 a 4, é feita uma breve introdução aos conceitos que serão utilizados ao longo do trabalho. No capítulo “Agentes Inteligentes” é feita uma abordagem aos agentes, sua evolução e características bem como uma breve introdução à Inteligência Artificial. No capítulo “Sistemas Multiagentes” é feita uma abordagem a estes sistemas e suas às características. No capítulo “Paradigmas da Inteligência Artificial” é feita uma abordagem e caracterização dos paradigmas ou metáforas utilizadas na Inteligência Artificial.

A segunda parte, composta dos capítulos 5 e 6, é abordada uma justificação natural do objetivo deste trabalho. No capítulo “Inteligência de Grupo” discute-se como o conhecimento é utilizado e partilhado em grupo. Este baseia-se na observação da natureza e a sua aplicação em diversos algoritmos que beneficiam desta. No capítulo “Conhecimento Coletivo” abordamos o conceito do conhecimento num grupo, com recurso a exemplos naturais. Depois abordamos a representação do conhecimento nos paradigmas da Inteligência Artificial - IA.

A terceira e última parte, composta pelos restantes capítulos, apresenta os elementos necessário para o suporte da proposta para o modelo. No capítulo “Representação do Conhecimento” discute-se a sua execução e diversas abordagens à sua representação. No capítulo “Dinâmica do Conhecimento” é feito um resumo das técnicas utilizadas para a revisão do conhecimento entre e intra-agentes. No capítulo “Teoria da Decisão” é feito um resumo deste campo e como pode ser aplicado no conhecimento coletivo.

No capítulo 10 são apresentadas bases um modelo para a representação do conhecimento coletivo no paradigma simbólico.

Nas conclusões discute-se as observações realizadas e que desenvolvimentos podem ser efetuados.

Parte I - Introdução

Breve introdução à Inteligência Artificial e à sua história. Agentes Inteligentes e Sistemas Multiagentes. Paradigmas da Inteligência Artificial.

2. AGENTES INTELIGENTES

O conceito de Inteligência Artificial tem um peso nas emoções das pessoas com quem se discute o assunto. Geralmente o primeiro ponto a ser levantado é “O que é a Inteligência.”. Ora só este ponto é muito complexo, pois, provavelmente, existem definições de inteligência quantas as pessoas inqueridas sobre a mesma. Basta ver o trabalho de Gardner [01] a respeito da existência de múltiplas inteligências. E apenas estamos no estado biológico, ou seja, aferir a inteligência a partir da observação e comportamento dos seres vivos, não entramos ainda no estado tecnológico. Todos nós já observamos a ação de algum animal e pensamos para nós mesmo “é mesmo inteligente”. Então o que é que nós observamos que nos dá a sensação de inteligência? Talvez o primeiro facto é a tentação de dotar os elementos à nossa volta de emoções humanas. Começando nos primórdios da existência humana a explicação para os fenómenos que rodeavam os humanos foram antropomorfizados para que se conseguisse dar um sentido à vida em geral. A nossa medida de inteligência tem a ver com a capacidade de o elemento observado mostrar características que reconhecemos como humanas. O conceito de agente inteligente não é novo na IA, mas tem ganho proeminência através do trabalho de Russel e Norvig [02]. Um agente refere-se a um sistema que geralmente processa informação recebida e produz um resultado. Este resultado depende do tipo de informação e do tipo de agente.

2.1. INTELIGÊNCIA ARTIFICIAL

Inteligência é definida, no dicionário, como a capacidade para aprender, raciocinar, compreender e outras formas de atividade mental. Aptidão em entender verdades, relações, factos, sentidos, intenções entre outros. Geralmente consideramos a inteligência como uma qualidade dos seres humanos de serem capazes de resolver problemas e pensar de forma racional. A IA é, portanto, a qualidade de um objeto que aparenta, ao observador, ter um comportamento inteligente ou parecido a um ser humano. Pode ser considerada como uma ciência, que procura estudar e compreender que é a inteligência, e, por outro lado, um ramo da engenharia que procura construir formas de simular o pensamento e fenómenos cognitivos. A IA abrange um número de campos que suportam a seu desenvolvimento, Tabela 2.1. A Figura 2.1 mostra uma relação entre os antecedentes da IA e a sua aplicação.

Tabela 2.1 – Inteligência Artificial e campos relacionados

Filosofia	Lógica, métodos de raciocínio, linguagem, racionalidade, aprendizagem
Matemática	Representação formal e provas, algoritmos, computação, (in)decidibilidade
Estatística	Modelação de incerteza, estudo dos dados
Economia	Teoria da decisão, agentes económicos racionais
Neurociência	Processamento de informação nos neurónios
Psicologia	Comportamento, perceção, processo cognitivo
Computação	Construção de computadores mais rápidos
Teoria Controlo	Sistemas de maximização de objetivos
Linguística	Representação de conhecimento, gramática

Também por vezes damos intenções e emoções aos objetos. Quando um aparelho não funciona como o esperado por vezes indignamos-nos pela sua falta de cooperação. Aplicamos as emoções humanas a objetos e/ou à natureza como nos podemos lembrar dos diversos mitos existentes.

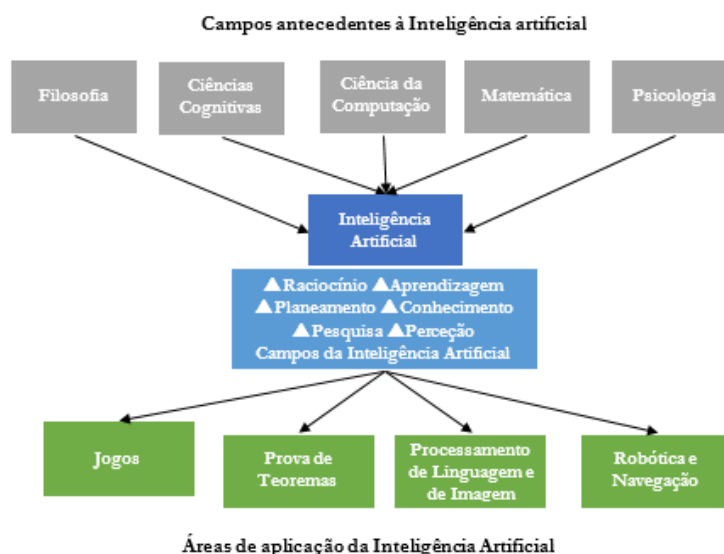


Figura 2.1 – Inteligência Artificial, campos antecedentes e aplicações

Estes dão-nos uma ideia do que a humanidade pensava sobre a inteligência dos elementos. Da história vem-nos os seres artificiais do mito de **Hephaestus**. Considerado o deus grego da tecnologia, dos ferreiros e artesãos entre as suas criações destaca-se os robots que construiu para o ajudar e **Talos** um homem de bronze. E **Galatea** de **Pygmalion** de Ovídeo. Passando a elementos mais reais temos **Aristóteles** que nos trouxe a lógica silogística o primeiro sistema de lógica formal dedutiva no quinto século A.C.. Na Idade Média apareceram as cabeças falantes criadas por **Gerbert de Aurillac**, Papa Silvestre II, (945-1003), **Albertus Magnus** (1198-1280) e o cientista e monge inglês **Roger Bacon** (1214-1294). O mecanismo para a sua operação perdeu-se e não é claro o seu funcionamento [03]. **Ramon Lull**, teologista espanhol, criou máquinas para desvendar verdades não matemáticas através da combinação de verdades elementares, criando, essencialmente, um primeiro sistema de lógica [04]. A criação da máquina de impressão por **Gutenberg** (1456), relógios (séc. XV-XVI) e a criação de animais mecânicos e outros instrumentos. No séc. 17 **Descartes** propõe que os animais não são mais que máquinas complexas. **Hobbes** publicou “*The Leviathan*” que contém uma teoria combinatoria do pensamento. **Pascal** cria primeira máquina mecânica de cálculo (1642). **Leibniz** melhora a máquina de Pascal (multiplicação e divisão) e prevê um cálculo universal de raciocínio no qual os argumentos poderiam ser decididos mecanicamente. No séc. XVIII aparecem os brinquedos mecânicos – o pato de **Vaucanson** e o jogador de xadrez de **Von Kempelen**. No séc. XIX **George Boole** desenvolveu a álgebra binária publicada no “*The Laws of Thought*”. **Charles Babbage** e **Ada Byron** (Lady Lovelace) trabalharam em máquinas calculadoras mecânicas programáveis. **Mary Shelley** publica a história “*Frankenstein*” em 1818. **Edward Lee Thorndike** expôs o Behaviorismo em “*Animal Intelligence*”. Estes elementos apresentam uma espécie de pré-história da inteligência artificial na qual observamos o surgimento de alguns dos temas abordados. No início do séc. XX deixamos a pré-história para

entrar num campo já mais definido. **Bertrand Russel** e **Alfred North Whitehead** publicam “*Principia Mathematica*” (três volumes publicados em 1910, 1912 e 1913; em 1927 a segunda edição) em que apresentam uma tentativa de descrição de um conjunto de axiomas e regras de dedução/inferência na lógica simbólica. Produzida em 1921 a peça “*Rossum’s Universal Robots*” (Rosumovi Univerzální Roboti) por **Karel Čapek** apresenta pela primeira vez a palavra robot [05]. Esta é derivada da palavra checa/eslava *robot* para trabalhador/servidão/trabalho forçado. Na peça os robots são construídos para substituir os trabalhadores de uma fábrica. Em 1926 **Fritz Lang** introduz os robots no cinema com o filme “*Metropolis*”. Uma tendência que continua nos dias de hoje quer no cinema quer na televisão ou noutra meio de entretenimento. **Warren S. McCulloch** e **Walter H. Pitts** publicam “*A Logical Calculus of the Ideas Immanent in Nervous Activity*” em 1943 criando as fundações das redes neuronais. Em 1943 o termo cibernética aparece por **Arturo Rosenblueth Stearns**, **Norbert Winer** e **Julian Bigelow** através da publicação de “*Behavior, Purpose and Teleology*”. Em 1945 **Vannevar Bush** publica “*As We May Think*” onde antevê um futuro em que os computadores assistem os humanos em diversas atividades. As bases ou fundações da Inteligência Artificial estão maduras sendo que apenas falta um passo para que seja mais tangível.

Esse passo pode ser alcançado através de alguns elementos apresentados nos anos 40 do séc. XX. Em 1945 o ENIAC e em 1949 o EDVAC o primeiro computador eletrônico digital e o primeiro programa de computador armazenado respetivamente. Outro é o teste de *Turing* onde se testa a capacidade de uma máquina apresentar um comportamento inteligente como, ou indistinguível de um ser humano. Este foi introduzido por **Alan Turing** no artigo “*Computing Machinery and Intelligence*”. Neste perguntava-se se as máquinas pensavam e Turing avançava que se o comportamento de uma máquina conseguisse iludir um humano da sua realidade então esta seria considerada inteligente. Um elemento importante seria a *Conferência de Dartmouth* organizada por **John McCarthy**, formalmente proposta por este **Marvin Minsky**, **Nathaniel Rochester** e **Claude Shannon** em 1956. Este encontro debruçou-se sobre a possibilidade de cada aspeto da aprendizagem e cada característica da inteligência serem precisamente descritas. Se era possível então podiam-se criar máquinas que as simulassem [06]. Dessa forma chegou-se ao período denominado de “Anos Dourados” entre 1956 e 1974. Nos anos 60 o financiamento para os centros de investigação de IA aumenta, construção de autómatos inteligentes e pesquisa em espaços de procura complexa. Aparecem os primeiros programas funcionais: jogo de dama (que aprende) de **Arthur Samuel**; **Allen Newell**, **Herbert Simon** e **Cliff Shaw** criaram o programa “*Logic Theorist*” para simular o processo de resolução de problemas humano. Iria provar 38 dos primeiros 52 teoremas apresentados no “*Principia Mathematica*” [07]. **Herbert Gelernter** apresentou uma máquina de prova de teoremas geométricos. **John Alan Robinson** completou o algoritmo automático de prova de teoremas. Em 1958 **John McCarthy** criou a linguagem de programação para IA – LISP [08]. Entre 1964 e 1966 **Joseph Weizenbaum** desenvolveu o primeiro *chatbot*, ELIZA, que conseguia desenvolver uma conversação realística, e até questionar algumas das expressões apresentadas. Depois deste crescimento na área surge um

período de retração. Este surge pelas dúvidas sobre a viabilidade da abordagem tomada. Além destas dúvidas alguns problemas surgem: a limitada capacidade de computação, tempo exponencial de resolução, senso comum e raciocínio, paradoxo de **Moravec**, o argumento do quarto chinês e redução de funcionamento. O paradoxo de **Moravec** refere que, contrário ao assumido, o raciocínio de alto nível necessita de pouca computação, mas o processamento dos sensores/atuadores exigem grande quantidade de poder de computação. Este foi apresentado por **Hans Moravec**, **Rodney Brooks**, **Marvin Minsky** e outros na década de 80. O argumento do quarto chinês, proposto por **John Searle** em 1980, descreve uma pessoa num quarto que só sabe Inglês. Recebe apontamentos, do exterior, com símbolos em chinês e acompanhado com um livro de regras sobre a correlação entre os diversos símbolos em chinês consegue realizar algumas instruções com os símbolos e enviar uma resposta novamente com símbolos chineses. Assim é possível que a pessoa do exterior faça questões em chinês, receba as respostas em chinês e fique convencida que a pessoa no interior do quarto também o compreende. **Searle** defende que o quarto, sistema, não tem inteligência. Estes pontos foram alguns daqueles que deixaram a IA com espaço de manobra mais reduzido.

Mas alguns elementos continuaram a destacar-se: os sistemas periciais. Em 1972 o **MYCIN** diagnostica doenças sanguíneas a partir de um conjunto de 600 regras e perguntas formuladas por si (por vezes melhores que os especialistas) [09]. O **XCON** (1980) um sistema de produção baseada em regras que assistia na conceção do tipo de computador fazendo as encomendas de partes a partir dos requisitos dos clientes. Utilizava 2500 regras para processar 80000 pedidos com 95%-98% de eficiência [10]. Em 1981 **PROSPECTOR** assistia na exploração mineira ao recomendar locais para exploração preliminar. Nos anos 90 aparece o primeiro *browser*, *Mosaic* pela **NCSA**, o computador **Deep Blue** vence **Gary Kasparov** em xadrez, aparece a **RoboCup** com jogadores robóticos. A **Sony** apresenta o **AIBO** o cão robótico, um agente dirige autonomamente a sonda **Deep Space 1**. A partir do início deste século a velocidade de desenvolvimento acelera. A **iRobot** apresenta o aspirador robótico **Roomba**, A **DARPA** introduz o **DARPA Grand Challenge** em que automóveis sem condutores disputam uma corrida. Continuam a aparecer novas aplicações: o projeto **Robotics Cog** para a construção de um robot humanoide, o **Loebner Prize** que desde 1990 premeia o melhor computador que se parece com um ser humano ao nível do processamento de linguagem (uma implementação do teste de **Turing**), o robot **ASIMO** produzido pela Honda e **Data Mining** entre outras aplicações. O *Data Mining* torna-se importante pela capacidade de processamento que é cada vez maior. Permitindo às empresas gerar grande quantidade de informação sobre o seu negócio e processá-la para obter informação relevante para os seus interesses. A modelação da bolsa de valores, projeto **LETS** [11], e o diagnóstico médico apresentam resultados relevantes [12] [13].

2.2. AGENTES

Podemos considerar dois tipos de agentes: por *software* ou *hardware*. Dessa forma temos dois campos de estudo com as suas características próprias. Os agentes de *software* que são que irão calcular um

resultado a partir da informação entrada e na robótica os agentes serão robots que terão sensores e atuadores para perceber e interagir com o mundo. Interessa então definir um agente. Seguindo [14] um agente é um sistema que está situado num ambiente e que é capaz de ação autónoma nesse mesmo ambiente para atingir os seus objetivos. Podemos observar a Tabela 2.2 com várias definições.

Tabela 2.2 – Definições de Agentes

<p>Russel & Norvig 2004 Um agente é qualquer entidade que percebe seu ambiente e através de sensores atua sobre esse mediante atuadores. Um agente é racional quando realiza a menor ação possível a partir dos dados percebidos.</p>	<p>Wooldridge 2004 Um agente inteligente é um sistema (hardware ou software) situado em um determinado ambiente, capaz de atuar de forma autónoma e racional dentro do ambiente para executar seus objetivos propostos</p>
<p>River 1996 Um agente é um programa de computador que funciona em <i>background</i> e desenvolve tarefas autónomas conforme delegadas pelo utilizador.</p>	<p>Michael Coen 1996 Agentes são programas que desenvolvem diálogos, negociam e coordenam transferência de informações.</p>
<p>Virdhagriswaran 1995 Os agentes apresentam conceitos de habilidade para execução autónoma e habilidade para executar raciocínio orientado ao domínio.</p>	

Das definições apresentadas existem vários pontos comuns: o agente atua num ambiente a partir da informação recebida e de acordo com os seus objetivos. Assim temos de enquadrar os elementos descritos: o sistema, o ambiente, a ação e os objetivos pois irão influenciar o agente. O ambiente em que este está integrado irá influenciar o seu tipo (sistema) que irá melhor interagir com o ambiente. Pode ser como visto como um agente de *software* – programa que realiza determinadas ações como por exemplo um filtro de *spam* – ou de *hardware* – um *robot* num labirinto. Como se percebe o agente terá de ser elaborado de maneiras distintas. O *software* irá receber informações e tratá-las como qualquer programa de computador ou interagindo diretamente com o utilizador ou em segundo plano. Já um *robot* terá de receber a informação através sensores e agir através de atuadores. Um modelo de um agente genérico está representado na Figura 2.2.

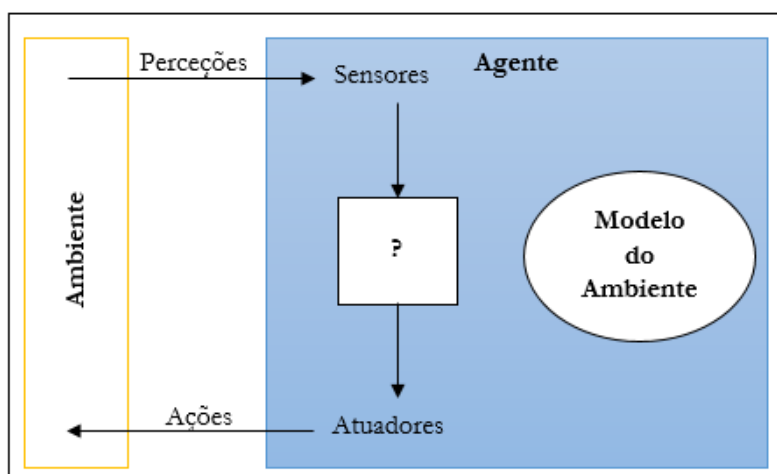


Figura 2.2 – Agente Genérico

A ação e os objetivos estão interligados pois as ações do agente terão em conta os objetivos a atingir. O que nos leva a um ponto importante que é como o agente atinge os seus objetivos. Isto irá depender da forma como o agente trata a informação e qual o seu nível de autonomia.

2.2.1. PROPRIEDADES

Existem duas noções de agente: fraca e forte [02]. A primeira, a mais aceite e abrangente, podendo ser considerada a generalização do termo. Desta advêm as seguintes propriedades:

- **Autonomia:** o agente opera sem intervenção externa e tem controlo sobre as suas ações e estado interno;
- **Habilidade Social:** o agente interage com outros elementos;
- **Reatividade:** o agente apercebe-se do ambiente e reage ao mesmo;
- **Pro-ação:** apesar da reatividade o agente procura agir de acordo com os seus objetivos.

A noção forte é influenciada pela inteligência artificial que pretende inculcar aos agentes algumas capacidades humanas. Para essa noção as seguintes propriedades são necessárias:

- **Conhecimento:** o agente tem capacidade de coletar informação para modelar o mundo em que se insere;
- **Crença:** o agente segue determinadas regras definidas que confia absolutamente;
- **Intenção:** o agente planifica as suas ações futuras;
- **Obrigaçãõ:** o agente compromete-se a cumprir os objetivos determinados;
- **Emoção:** características intrinsecamente humanas que são alvo de tentativa de aplicação nos agentes.

Outras propriedades podem ser utilizadas para caracterizar os agentes. Estas, por vezes, dependem do sistema. Por exemplo:

- **Mobilidade:** a capacidade do agente se deslocar no seu ambiente;
- **Veracidade:** o agente não veicula informação errada de forma deliberada;
- **Benevolência:** o agente poderá cumprir objetivos de outros agentes que não comprometam os seus;
- **Racionalidade:** os agentes agem sempre de forma a cumprir os seus objetivos.

Será simples de encontrar contextos para que as propriedades tenham relevância para o agente.

2.2.2. ARQUITETURAS

Uma arquitetura de um agente pode ser descrita como a forma da sua construção. A metodologia utilizada na sua construção permite compreender como será possível decompor o agente em vários módulos e como estes se relacionam [15]. Podemos descrever três arquiteturas principais: deliberativa, reativa e híbrida.

2.2.2.1. ARQUITETURA DELIBERATIVA

Baseada no modelo simbólico do mundo do agente. Este toma as decisões manipulando os símbolos. Nesta é necessário representar a informação do mundo real na forma de um modelo. A Figura 2.3 apresenta o modelo do agente.

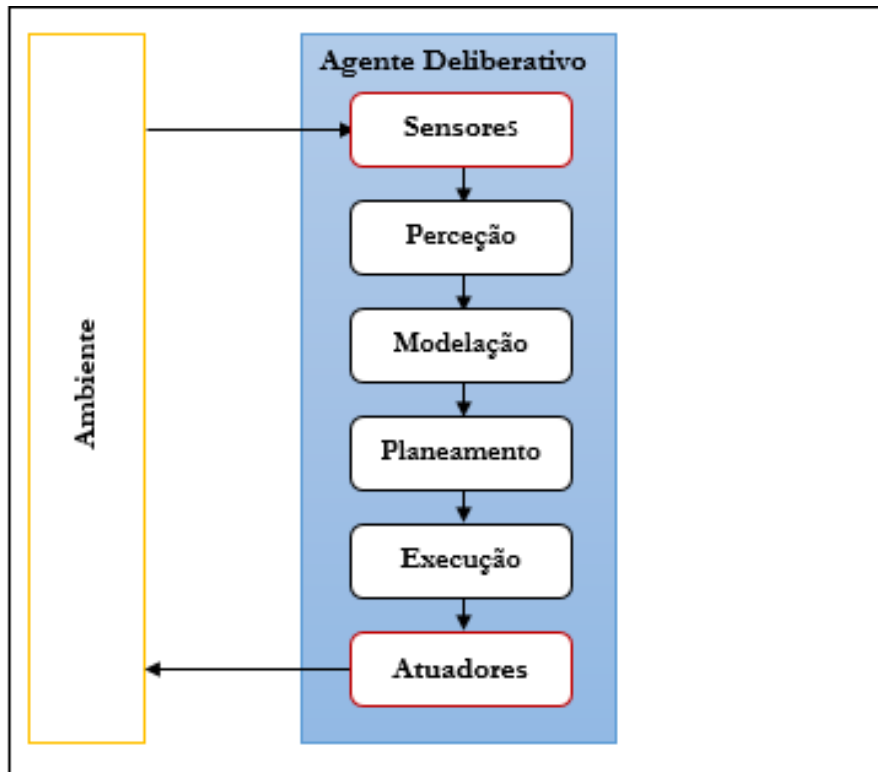


Figura 2.3 – Arquitetura Deliberativa

2.2.2.2. ARQUITETURA REATIVA

O agente baseia-se num conjunto simples de regras (condição \rightarrow ação). Não necessita de uma representação do ambiente e não apresenta processo de raciocínio. A Figura 2.4 apresenta este modelo.

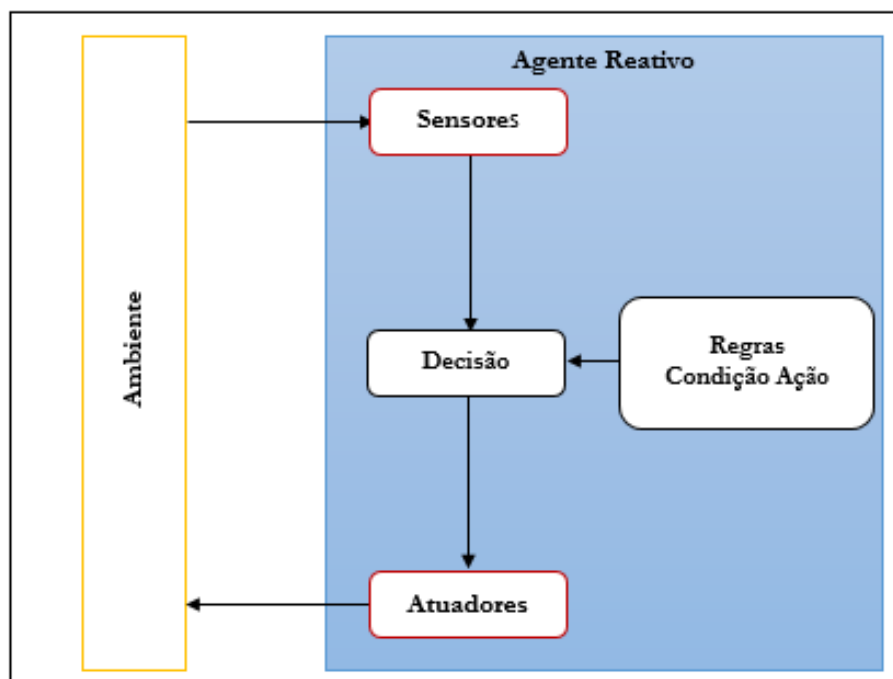


Figura 2.4 – Arquitetura Reativa

2.2.2.3. ARQUITETURA HÍBRIDA

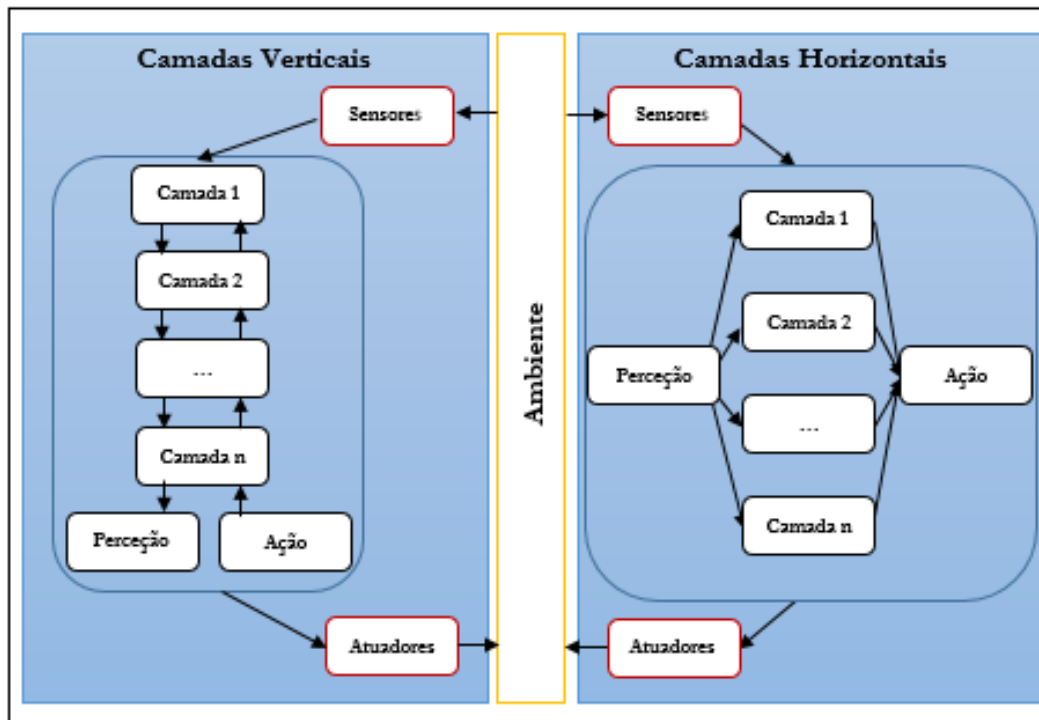


Figura 2.5 – Arquitetura Híbrida

A Figura 2.5 apresenta o modelo da arquitetura que tenta reunir o melhor das duas anteriores. Assim necessita de combinar a representação simbólica do ambiente, com a ação reativa para responder a estímulos externos. Sem comprometer o processo de raciocínio. Geralmente é conseguido através de uma arquitetura em camadas, estas estão organizadas por nível de abstração. As camadas superiores apresentam uma maior abstração. A organização destas camadas pode ser de forma horizontal ou vertical. Na forma horizontal todas as camadas têm contato com o ambiente contrariamente à organização vertical em que apenas as primeiras camadas mantêm o contato com o ambiente.

2.3. CONCLUSÃO

Neste capítulo fizemos a introdução do conceito de agente inteligente. Apresentamos brevemente o conceito de inteligência que apresenta várias interpretações, mas o comum parece centrar-se na capacidade de realizar processos mentais tomar decisões. Além de uma abordagem, também breve, à evolução desta. Da mesma forma a IA será a qualidade de uma agente ser capaz de suportar tais processos. Não num nível natural, mas sim simulado. Esta simulação será conforme o agente em concreto: *software* ou *hardware*. O agente em *software* existe nos sistemas computacionais e o agente em *hardware* será um robot. A partir deste princípio exploramos propriedades desejáveis e arquiteturas de um agente.

3. SISTEMAS MULTIAGENTES

É um campo relativamente recente que nasceu do campo de IA Distribuída pertencente à Inteligência Artificial dedicada ao desenvolvimento de soluções distribuídas para resolução de problemas que requerem inteligência. Estes são desenvolvidos para que coordenem o seu conhecimento, capacidade e objetivos para que possam executar ações ou resolver problemas. Os agentes podem trabalhar em conjunto, para atingirem o mesmo objetivo comum ou individualmente para o seu objetivo. É uma área em grande expansão. A criação da agência Europeia *AgentLink* [16] para a coordenação desta área e a competição internacional de futebol *Robocup* demonstram o seu âmbito. Estes sistemas são formados por vários agentes como duas propriedades importantes [14]:

- **Autonomia:** os agentes necessitam de decidir por si atingir os seus objetivos;
- **Interatividade:** os agentes necessitam de interagir com outros agentes para troca de informação, cooperação ou negociação.

A grande motivação dos Sistemas Multiagentes, SMAs, são problemas que necessitam de resolução de forma distribuída. É possível dividir um problema entre vários agentes tornando o mesmo mais simples. Dessa forma ao invés de considerarmos um robot com todas as implicações que podem surgir; consideramos minirobots, ou agentes, que com pouco custo e trabalhando em conjunto podem efetuar o trabalho de um robot mais complexo e de maior custo. Devido à contínua redução dos custos envolvidos na computação é possível a introdução dos microprocessadores em cada vez mais locais. Além desta ubiquidade verifica-se que os sistemas funcionam cada vez mais em conjunto ao invés de funcionamento independente e, estes, cada vez mais realizam as nossas tarefas. Depois torna-se claro que o passo seguinte será a introdução de inteligência nestes sistemas. Assim temos vários sistemas a cooperar entre si com a inteligência suficiente para tomar as melhores decisões. Estes conceitos podem ser levados para além dos sistemas informáticos convencionais. Ao utilizar os robots, agentes, a cooperar podem realizar as tarefas que lhes são incumbidas tão bem como apenas um robot isolado. Ao realizar esta conceção há duas perspetivas a ter em consideração: macro e micro. A micro considera o desenho do agente que seja capaz de ação independente e autónoma, a macro considera o desenho da sociedade onde os agentes se encontram, mais precisamente a sua interação. Um agente é um sistema computacional capaz de ação independente num determinado ambiente. Para isso é necessário que possa obter informações sobre o ambiente que o rodeia para que possa tomar uma melhor decisão. Podemos considerar vários desses agentes como o termóstato, que mantém a temperatura ao nível pedido. Claro que queremos agentes que possam tomar a iniciativa de forma a chegar ao objetivo. Considerando vários agentes temos que verificar como vão interagir, atendendo que nem todos podem ter os mesmos objetivos. E considerar uma forma de comunicação entre todos. Além de considerar sobre que ambiente irão atuar. Observamos que a forma de desenhar estes agentes desvia-se do convencional pois temos de considerar metáforas que descrevem melhor a perceção do ambiente em vez da visão orientada à máquina. O que provoca uma

procura de metáforas que consigam explicar o funcionamento dos sistemas, cada vez são mais complexos. Tendo em conta todos estes aspetos a criação de uma sociedade robótica permanece ainda distante. Para a criação de uma verdadeira sociedade existe um aspeto crucial: a comunicação. Sem esta não será possível qualquer aproximação a esta. Pois o indivíduo iria atuar sozinho sem se preocupar com outros dentro do seu ambiente. Assim é necessário perceber como a comunicação é feita e como afeta a percepção do ambiente. Os agentes terão, como fazem as pessoas, de negociar, de ceder, de exigir para atingir os seus objetivos bem como chegar a uma solução que satisfaça a todos se não for possível a obtenção dos objetivos individuais. Numa análise global os sistemas computacionais existentes estão permanentemente a interagir, e a caminhar para interação cada vez maior. A existência destes conceitos permite, não só a construção de robots que interagem, mas também possibilidade de estudar comportamentos sociais e computacionais. Este sistema é composto de vários agentes e como cada é capaz de interação com os outros cria-se uma comunidade com comportamento social capaz de criar inteligência e conhecimento. Como cada vez mais a interligação faz parte do dia-a-dia a interação entre os sistemas é um ponto de grande importância senão mesmo crítico. No início foram adotadas duas estratégias: *Blackboard Systems* e *ACTOR Model*. Na primeira existe um conhecimento partilhado comum a todos, como os antigos quadros das escolas. O segundo elimina a memória comum a favor da interação entre os agentes. Assim sendo os agentes são capazes de reagir à informação recebida e agir com base na mesma. Estes podem ser vistos de duas perspetivas distintas consoante o seu comportamento: reativo ou cognitivo. O agente reativo recebe a informação e age apenas em resultado da mesma sem ter em conta ações passadas ou previsões sobre ações futuras, ou seja, comporta-se sem memória. O agente cognitivo pode ser visto como um sistema intencional com algum tipo de memória e capaz de manipular conhecimento.

Os SMAs pretendem resolver problemas através do comportamento autónomo de agentes em interação que seriam demasiado complexos de resolver por um agente isolado. As vantagens de um SMA sobre um sistema central podem ser resumidas da seguinte forma:

- Maior velocidade de resolução pelo paralelismo de todos os agentes;
- Menor custo de comunicação ao substituir o conjunto de informação num ponto central pela transmissão de informação tratada entre os agentes;
- Maior flexibilidade pela utilização de grupos de agentes com diferentes características na resolução de problemas;
- Maior segurança/redundância com novos agentes a tomar lugar de agentes que se perdem.

3.1. COMUNICAÇÃO

A comunicação é um aspeto crucial na interação ente agentes. As propriedades *Habilidade Social* e *Interatividade* estão intimamente relacionadas com a comunicação, sendo, nos SMAs, a comunicação tratada a alto nível, ou seja, próximo da linguagem humana. A Teoria do Ato do Discurso, *Speech Act Theory* (1962), é um ponto importante do estudo da Comunicação. John Austin forneceu uma importante observação ao afirmar que a comunicação é uma forma de ação. A afirmação de algo dá

existência ao mesmo. Austin apontou que toda a comunicação podia ser fraseada na forma declarativa com a utilização de verbos performativos. Assim a comunicação passa a ser uma sequência de informações, diretivas e promessas. A estilização da comunicação promove uma construção interessante para os SMA. Os agentes necessitam de um módulo para a comunicação para que possam receber e/ou enviar mensagens, este é relacionado com o tipo de agente. A comunicação exhibe características relacionadas com a linguagem utilizada pelos agentes [17]. Estas deverão ser tidas em conta: sintaxe, semântica, vocabulário, pragmática e modelo do domínio do discurso.

Na IA é presumido que os agentes são construídos com base e, conceitos cognitivos como crenças, objetivos e intenções. A comunicação de tais agentes é especificada em termos da sua representação cognitiva. Os pontos de partida foram a interação humano-computador e a linguagem natural que inclui os temas de compreensão do discurso ou texto e compreensão da fala. O ponto comum destas abordagens é o desenvolvimento que se deveu à descoberta duma ferramenta que assistisse um utilizador a obter informação de uma base de dados ou a executar tarefas simples como obter um bilhete. A funcionalidade chave de tais ferramentas era inferir a tarefa que o utilizador desejava realizar e apoiá-lo na sua execução. Estas mantinham um modelo do utilizador e eram configuradas com um modelo do domínio sobre o qual raciocinam através de heurísticas para determinar o melhor processo de resposta às expectativas dos utilizadores e, até, antecipar os pedidos destes. Estas não podiam falhar na sua cooperação com o utilizador e, além disso, o utilizador era considerado como cooperante. A ferramenta não concebia que o utilizador ia, propositadamente, ser enganador nas suas intenções. Estas desenvolveram-se e tornaram-se mais pró-ativas, sendo vistas como agentes que, em alguns casos, comunicavam entre si e não só com o seu utilizador. Estas mantinham o modelo do seu utilizador e de outros através da comunicação trocados com outras ferramentas. Podiam fazer inferências das crenças e intenções de outros e agir, e comunicar, de acordo com as necessidades. Este funcionamento pareceu ser indicado para a comunicação entre agentes. A outra ideia da comunicação surgiu da utilização de sistemas de conhecimento distribuído (sistemas expertos com capacidade de comunicação entre eles). Cada agente incluía um raciocinador e representação do conhecimento e a comunicação era um meio de partilhar tal informação. As suas assunções da comunicação: os agentes eram construídos com a mesma representação do conhecimento e os agentes eram cooperativos. As linguagens mais utilizadas foram o KQML – *Knowledge Query Manipulation Language*, ACL – *Agent Communication Language* e ICL – *Inter-Agent Communication Language*.

3.2. COOPERAÇÃO/COMPETIÇÃO

Estes dois campos, competição e cooperação, são duas vertentes [17] do SMAs em que o comportamento dos agentes leva em conta a prioridade dos seus objetivos próprios ou da sua comunidade. Através da cooperação os agentes conseguem os seus objetivos de duas formas distintas: A primeira será um agente ter um objetivo concreto e através, não só das suas capacidades, das interações com outros agentes, neste caso cooperação, conseguir o seu objetivo individual. Outra

forma será o objetivo ser comum a todos os agentes e a sua cooperação ser voltada para a obtenção do objetivo comum a todos. Na competição podemos observar os mesmos dois conceitos: um objetivo individual ou comum. No primeiro caso um agente compete com outros pelo mesmo objetivo. Esta situação cria um funcionamento distinto ao anterior pois o agente procura atingir o seu objetivo primeiro, sem cooperação com outros. Embora possamos elaborar um cenário em que os agentes colaboram entre si para impedir que outros consigam o objetivo tentando facilitar o seu próprio sucesso. Noutro caso toda uma comunidade de agentes compete com uma outra comunidade para o seu objetivo comum.

3.3. PERSPETIVAS

Do discutido anteriormente existem duas perspetivas a considerar num SMA: perspetiva do agente e perspetiva do grupo ou comunidade. A primeira lida com:

- Categorias do agente;
- Estrutura e manutenção do conhecimento;
- Capacidade de raciocínio;
- Capacidade de adaptação e aprendizagem;
- Arquiteturas do agente.

Já a segunda:

- Organização;
- Coordenação;
- Cooperação;
- Negociação;
- Coerência de comportamento;
- Planeamento;
- Comunicação;
- Interação.

Com se nota a primeira perspetiva trata de promover a correta definição do agente que é pretendido. Definindo as suas propriedades e funções tendo em conta a sua utilização. Um agente que centra o seu funcionamento na capacidade de cooperação provavelmente terá uma definição diferente de um agente competitivo. Também será importante perceber o ambiente onde o agente irá agir para melhor definição das suas propriedades. A segunda perspetiva eleva a fasquia pois trata de definir uma comunidade. Simplesmente observando as comunidades existentes no mundo natural compreende-se que é uma perspetiva complexa e de grande importância. Pois temos de lidar com vários agentes conseguir a sua interação, quer para cooperar quer para competir, num determinado ambiente com um determinado objetivo. Estas duas perspetivas acabam por se influenciar pois não podem ser tratadas de forma totalmente independente.

3.4. CONCLUSÃO

Neste capítulo discutimos os SMAs. Estes trazem mais um ponto importante: possibilidade de uma sociedade de agentes que interagem. Essa interação é de grande importância pois levanta uma questão pertinente. O que é produzido durante a mesma? No capítulo vimos que os agentes podem ter um objetivo a atingir e para o atingir podem cooperar ou competir. Esta cooperação envolve troca de informação. Esta troca de informação levará a um conhecimento de grupo.

4. PARADIGMAS INTELIGÊNCIA ARTIFICIAL

As aplicações da Inteligência Artificial englobam quatro requisitos para o seu desenvolvimento: pesquisa e otimização, representação do conhecimento, aprendizagem e planificação. No trabalho utilizaremos as definições de Russel e Norvig. A Tabela 4.1 mostra algumas das definições da IA de diferentes abordagens.

Tabela 4.1 – Definições de Inteligência Artificial

<p>Thinking Humanly “The exciting new effort to make computers think ... <i>machines with minds</i>, in the full and literal sense.” (Haugeland, 1985)</p> <p>“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ...” (Hellman, 1978)</p>	<p>Thinking Rationally “The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985)</p> <p>“The study of the computations that make it possible to perceive, reason, and act.” (Winston, 1992)</p>
<p>Acting Humanly “The art of creating machines that perform functions that require intelligence when performed by people.” (Kurzweil, 1990)</p> <p>“The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)</p>	<p>Acting Rationally “Computational Intelligence is the study of the design of intelligent agents.” (Poole et al, 1998)</p> <p>“AI ... is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)</p>

A finalidade da inteligência artificial, por Sage [18], é o desenvolvimento de paradigmas ou algoritmos para máquinas que realizam tarefas cognitivas nas quais os humanos apresentam uma grande vantagem. Esse sistema deve ser capaz de sustentar três tarefas primordiais:

- Armazenar conhecimento;
- Aplicar este conhecimento para a resolução de problemas;
- Adquirir novo conhecimento através da experiência.

Então o sistema apresenta três componentes fundamentais: representação, raciocínio e aprendizagem [19] como ilustrado na Figura 4.1.



Figura 4.1 – Componentes de um Sistema de Inteligência Artificial por Sage

Representação

Uma das mais distintivas características da IA será a utilização de uma linguagem e estruturas simbólicas para a representação do conhecimento sobre o domínio e o conhecimento específico da solução do problema. Os símbolos são normalmente apresentados de uma forma familiar o que os torna mais simples de serem entendidos por um humano. Esta simplicidade torna-a adequado para uma comunicação humano-computador. O conhecimento, tal como é entendido pelos investigadores da IA, é um termo aplicado aos dados a serem tratados. A sua representação pode ser declarativa ou procedimental. Na primeira o conhecimento é representado por uma coleção de factos estáticos com um conjunto de procedimentos gerais para a sua manipulação. Na segunda o conhecimento está incorporado num código executável que representa o significado do conhecimento.

Raciocínio

Na forma mais simplista é a capacidade de resolver problemas. Para que um determinado sistema seja considerado como qualificado como sistema de raciocínio tem de satisfazer determinadas condições [20]:

- Deve ser capaz de expressar e resolver um conjunto abrangente de problemas e tipos de problemas;
- Deve ser capaz de reconhecer a informação explícita e implícita;
- Deve ter um mecanismo de controlo que determina as operações a serem aplicadas na resolução de um determinado problema quando a solução foi obtida ou quando encerrar o tratamento do problema.

Esta resolução de problemas pode ser encarada como um problema de busca. Uma maneira de lidar com a busca é utilizar regras, dados e controlo. As regras atuam sobre os dados e o controlo sobre as regras. Em algumas situações o conhecimento disponível é incompleto ou inexato. Nesta situação podem ser utilizados procedimentos de raciocínio probabilísticos para que os sistemas lidem com a incerteza [19].

Aprendizagem

Um modelo simples de aprendizagem, Figura 4.2, apresentado o ambiente fornece alguma informação para um agente com elementos de aprendizagem.



Figura 4.2 – Modelo simples de aprendizagem [19]

O elemento de aprendizagem utiliza esta informação para aperfeiçoar a base de conhecimento e o elemento de desempenho utiliza esta para realizar as suas tarefas. Geralmente a informação que o ambiente fornece não é perfeita pelo que o elemento de desempenho não sabe como preencher os detalhes ausente ou ignorar os menos importantes. Assim sendo a máquina, inicialmente, opera por suposição só depois recebe a realimentação do elemento de desempenho permitindo que faça uma

reavaliação, se necessário, das suas hipóteses. Esta aprendizagem utiliza dois métodos: indutivo e dedutivo. O processamento de informação indutivo revela padrões gerais e regras a partir dos dados e da experiência. O processamento dedutivo, por sua vez, utiliza regras gerais para determinar dados específicos. A importância das bases de conhecimento e as dificuldades de aprendizagem leva ao desenvolvimento de vários métodos para o seu aperfeiçoamento. A ideia dos sistemas especialistas é que é mais simples obter a experiência já compilada pelos especialistas de um determinado campo do que duplicar o caminho que os levou à sua competência.

4.1. PARADIGMA CONEXIONISTA

Este vê a inteligência como uma propriedade emergente das interações de unidades elementares de processamento [21]. Esta abordagem genérica deixa em aberto a possibilidade de uma grande abrangência de ideias. O sistema nervoso pode ser considerado em três estágios de um sistema, a Figura 4.3 apresenta este modelo. O centro do sistema é o cérebro que recebe informação, apreende-a e toma as decisões necessárias. A informação circula no sistema: os recetores convertem os estímulos do corpo humano ou do ambiente em impulsos elétricos que enviam a informação para o cérebro e os atuadores convertem os impulsos elétricos gerados pelo cérebro em respostas do sistema.



Figura 4.3 – Diagrama do Sistema Nervoso

O trabalho de Ramón y Cajál, em 1911, introduziu a ideia dos neurónios como os principais constituintes permitiu simplificar o estudo do cérebro. O neurónio é a base básica do funcionamento cerebral, como célula nervosa, a sua especificidade permite gerar sinais elétricos e transmiti-los a outros neurónios. Estes possuem uma membrana que os separa do seu meio, permitindo que dentro do seu espaço se mantenham as condições ideais para as suas funções orgânicas. Esta membrana compõe o corpo do neurónio permitindo que as organelas celulares efetuem as suas funções, o axónio, prolongamento único e longo e as dendrites que são prolongamentos curtos e mais numerosos. O axónio é capaz de disparar um impulso nervoso e transmiti-lo a distância através da conexão com dendrites e outros neurónios através dos seus axónios colaterais. Esta conexão é chamada de sinapse, um contacto entre as membranas das células intervenientes com um espaço chamado de fenda sináptica. Esta, ao receber o impulso nervoso, liberta uma substância química produzida pelo neurónio denominada de neurotransmissor. Esta explicação da sinapse é simplista pois existem outros elementos importantes na sua estrutura. No ponto que mais interessa é que o impulso não é transmitido diretamente de um neurónio para outro, mas sofre uma codificação química através dos neurotransmissores e recetores presentes nas sinapses. Sendo esta tradução do sinal elétrico, química e elétrico novamente uma flexibilidade do sistema nervoso que irá permitir a formação da memória, aprendizagem e muitas outras propriedades do cérebro.

Os impulsos elétricos são gerados através da ação da membrana do neurónio que através de canais existentes na sua estrutura que permitem a passagem de iões, especialmente sódio e potássio. A abertura destes canais varia conforme o potencial eléctrico presente. Quando atinge determinado nível na membrana os canais referentes ao sódio abrem-se para permitir a entrada destes iões num processo passivo pois a quantidade destes fora da célula é maior. Como o sódio é carregado positivamente o potencial aumenta. Quando este atinge um determinado nível os canais respeitantes ao potássio abrem-se e os iões saem da célula tendo uma ação análoga ao sódio, mas baixando o nível do potencial. Esta variação da diferença de potencial será a base do impulso nervoso. Um diagrama do neurónio está presente na Figura 4.4.

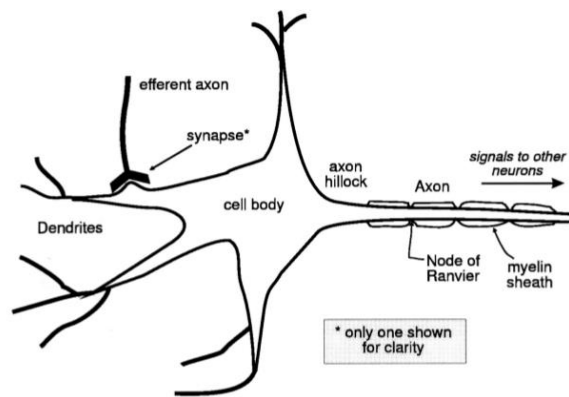


Figura 4.4 – Neurónio [22]

As redes neuronais foram motivadas pelo conhecimento de que o cérebro humano funciona de uma maneira distinta no processamento de informação relativamente ao computador digital. O cérebro é um computador, um sistema de processamento de informação, complexo, não linear e paralelo em que é capaz de se organizar para realizar tarefas de processamento complexas. Tais como reconhecimento de padrões, perceção, controlo motor, entre outras de maneira muito mais eficaz e eficiente do que os sistemas de computação. Warren McCulloch e Walter Pitts definiram o modelo matemático de um neurónio (1943), Figura 4.5, que seria análogo ao neurónio natural. É definido um conjunto de entradas com diferentes pesos que simulam a excitação das dendrites. Se a soma pesada ultrapassar um certo limite, de referência, faz disparar o neurónio para que produza um sinal. Existem várias interpretações dando azo a três aspetos de classificação possíveis: topológica (como os neurónios se interligam; função de ativação (ativação/desativação dos neurónios) e dinâmica de rede (a forma como os seus componentes variam).

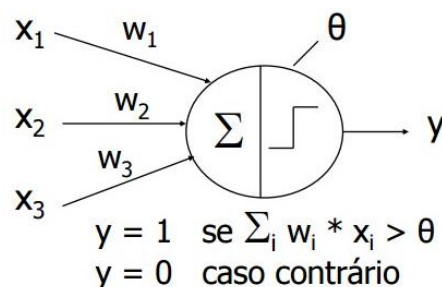


Figura 4.5 – Ativação de neurónio artificial segundo McCulloch e Pitts

Em 1982 John Hopfield baseou-se em modelos simples de neurónios e construiu uma rede neuronal com capacidade de simular a memória associativa humana acendendo através do conteúdo e não através de endereços como na computação [23]. A rede Hopfield, como ficou conhecido o seu trabalho, providencia um modelo para a compreensão da memória humana. Incorpora a modelação da memória associativa através da utilização de vetores de memória. Estes podem ser utilizados e irão despoletar a integração de vetores similares na rede. Este processo poderá encontrar intrusões. Na memória associativa da rede de Hopfield existem dois tipos de operações: auto-associação e hétero-associação. O primeiro indica a associação de vetor como si mesmo a segunda com a associação de dois vetores associados em memória. A função de ativação utilizada por Hopfield é não-linear. O que indica que a dinâmica modifica os valores dos vetores em direção a um dos padrões memorizados.

A rede neuronal aprende um modelo do ambiente em que se insere e tem de manter este modelo suficiente consistente com o mundo real para atingir os objetivos pretendidos. Este conhecimento consiste em dois tipos de informação:

- Estado conhecido do mundo, representado pelos factos sobre o que era conhecido-informação prévia;
- Observação do mundo (medidas) obtidas através de sensores para sondar o ambiente em que a rede neuronal se insere.

Nas redes neuronais os modelos de processamento paralelo distribuído são centrais. Estes assumem o processamento de informação através da interação de um grande número de neurónios, onde cada um envia determinados sinais – excitadores e/ou inibidores – para os restantes da rede [21]. O paralelismo é um conceito essencial ao processamento de informação em redes neuronais bem como o da flexibilidade. O paralelismo, que poderá ser extenso, possibilita às redes neuronais uma grande robustez. Pois a distribuição dos estados dos neurónios permite que aqueles que se desviem dos valores esperados, criando elementos ruidosos ou incompletos, não afetem o funcionamento da rede permitindo que funcione satisfatoriamente [19].

4.2. PARADIGMA EVOLUCIONISTA

Esta perspetiva encara a evolução, proposta por Darwin e complementada por Mendel, como uma solução viável dos problemas através da seleção natural adaptada a agentes computacionais.

A Natureza possui os meios de modificar os seres vivos em cada geração. As alterações são causadas pela alteração do código genético, quer seja por mutação aleatórias produzidas por radiação ionizantes, vírus ou mesmo ações químicas. Estes seres com código genético alterado acabam por se reproduzir com os existentes produzindo uma nova geração que incorpora, de alguma forma, as mutações ocorridas. O código genético armazena as instruções necessárias para o funcionamento das células através de uma coleção de genes organizados sequencialmente numa estrutura chamada cromossoma.

O gene é o componente básico dos seres vivos e estes possuem inúmeros genes organizados em cromossomas no núcleo de cada célula. A Figura 4.6 apresenta a relação entre uma célula e o gene.

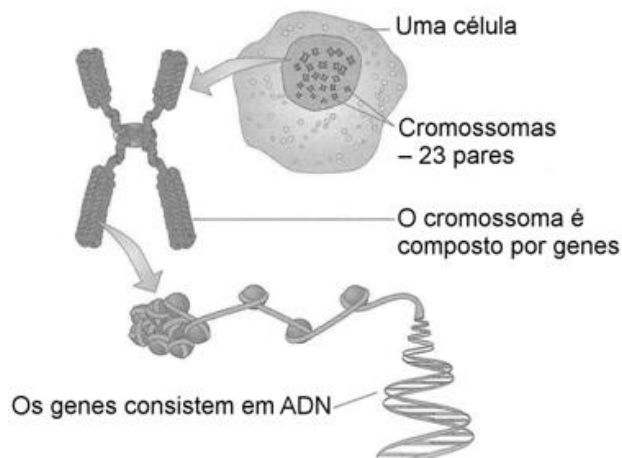


Figura 4.6 – Cromossoma e gene [24]

O material presente no código genético (genótipo) fornece as propriedades básicas ao ser vivo que, sob a influência do meio ambiente, acabam por gerar o ser vivo (fenótipo). No processo de mutação um gene é alterado aleatoriamente, criando um novo gene cuja expressão futura (fenótipo) não se pode prever *a priori*. Certas mutações irão levar à geração de fenótipos mais adaptados ao meio ambiente permitindo a continuidade da espécie diante doenças. Escassez de alimentos, clima e outros fatores. Outras mutações irão produzir seres vivos com fenótipos mal-adaptados que não sobrevirão às hostilidades do meio ambiente.

A reprodução permite que os cromossomas dos seres vivos se combinem definindo um novo ser vivo com novas características que irão ser postas à prova pela Natureza no período e local da sua vivência. No processo de reprodução, duas células haploides, com metade dos cromossomas da espécie, unem-se formando uma célula diploide contendo todos os cromossomas necessários à definição da espécie. Os cromossomas passam por uma fase de encadeamento, quebra e troca de material genético chamada “*crossing-over*” na qual o material genético de um dos seres vivos é combinado com o material de outro ser vivo. Os cromossomas resultantes serão, então, uma combinação dos genótipos dos seres envolvidos e um fenótipo novo será o resultado. [22]

Com base nestas observações os algoritmos genéticos seguem uma linha evolucionária semelhante fazendo que a solução de um problema evolua até satisfazer a condição final. Estes foram propostos por John Holland em 1975 em que a ideia, baseada na teoria de Darwin e Wallace, é de procurar a solução ótima para um problema a partir de uma população inicial, que representa um conjunto de soluções iniciais, candidatas à solução. A partir do emparelhamento destes indivíduos chegam uma nova geração. Desta seleciona-se os mais próximos da solução ótima, ou seja, os mais aptos, que irão conduzir a nova geração e assim por diante até chegar à solução.

4.3. PARADIGMA SIMBÓLICO

A inteligência como computação. Sendo a inteligência um conjunto de processos que atuam sobre estruturas simbólicas. Este sistema define três pontos necessários à sua aplicação: conhecimento, regras e sistema de controlo. O primeiro introduz a representação do conhecimento, ou seja, a sua representação simbólica. As regras permitem representar simbolicamente os operadores que são capazes de criar e alterar os símbolos que representam o conhecimento. Esta abordagem deve-se a Allen Newell e Herbert Simon que defenderam que os computadores e a mente humana são de uma família de artefactos denominados sistemas Físicos de Símbolos – SFS [22]. Então os SFS têm a capacidade de processar as estruturas simbólicas, a representação do conhecimento, ao longo do tempo criando novas estruturas. Os processos são: a criação, modificação, cópia ou destruição de símbolos. A inteligência é, então, o resultado da atuação dos processos sobre as estruturas simbólicas. Estes processos utilizam o raciocínio através de dedução, abdução, indução e analogia para a construção de estruturas simbólicas.

Na IA clássica é dado grande ênfase na construção de representações simbólicas – que são denominadas desta forma pois representam algo. A cognição da inteligência artificial assume a existência de representações mentais pelo que a modela como processamento sequencial de representações simbólicas. No processamento sequencial, como nos sistemas computacionais, as instruções são executadas passo a passo, mesmo quando não existe uma ordenação predeterminada. Esta terá vindo da natureza sequencial da linguagem natural e inferência lógica, bem como da estrutura da máquina de von Neumann, que precedeu a inteligência artificial clássica durante a mesma era intelectual [19].

A linguagem do pensamento pode ser considerada como um modelo para a IA clássica. As representações simbólicas possuem uma estrutura quase linguística. Assim como as expressões de linguagem natural estas podem ser complexas e construídas de forma sistemática a partir de símbolos simples. De um conjunto limitado de símbolos novas expressões podem ser compostas pela capacidade de composição das expressões simbólicas e da analogia entre a estrutura sintática e a semântica.

4.4. CONCLUSÃO

Neste capítulo discutimos brevemente o que é IA. Verificamos que não existe uma definição, mas sim várias perspetivas que apresentam várias propriedades, consideradas essenciais, para a inteligência. Como se esperaria, pois, a própria definição de inteligência encontra-se na mesma situação. De seguida observamos três paradigmas se se baseiam na observação do mundo natural. Estes paradigmas apresentam propriedades diversas que revelam diferentes aproximações e aplicações no campo da IA.

Parte II - Conhecimento Coletivo

Apresentação de alguns algoritmos que surgiram pela observação da natureza. Emergência do conhecimento através da interação dos elementos de um grupo. Representação do conhecimento nos paradigmas da Inteligência Artificial.

5. INTELIGÊNCIA DE GRUPO

Observando a natureza verificamos vários comportamentos e ações que permitem alguma inspiração. Uma inspiração é derivada pelo comportamento natural de vários organismos quando estes se organizam em grupo. Podemos observar o enxame de abelhas, formigas, colónias de bactérias ou bando de pássaros que no seu conjunto parecem mais inteligentes que os indivíduos pertencentes ao mesmo. Os indivíduos seguem regras simples e, mesmo, sem um centro de controlo de como devem agir acabam, pela sua interação, com um resultado inteligente. Este comportamento leva a uma inteligência emergente do comportamento global sem a consciência por parte do indivíduo. A inteligência de grupo pode ser explicada como o comportamento coletivo de sistemas descentralizados e autónomos. É um conceito utilizado na inteligência artificial que foi introduzido por Gerardo Beni e Jing Wan [25].

Existem vários algoritmos e sistemas com base nestas observações [26]:

- O transporte coletivo realizado por formigas inspirou o desenho de controladores de robots para a realização de trabalho coordenado;
- A ordenação de casulos das formigas motivou vários algoritmos de ordenação e *clustering*;
- As características de orientação e pesquisa de caminhos pela formiga *Catalyphis* foram utilizadas como arquétipo para a construção de uma unidade de orientação robótica;
- Modelos de divisão de trabalho entre membros de uma colónica de formigas foram utilizados para regular a divisão de tarefas entre robots.

Estas observações vêm de encontro ao anteriormente discutidos sobre SMAs. Elementos em cooperação conseguem melhores atingir objetivos mais complexos do que os possíveis individualmente.

5.1. BIOCOMPUTAÇÃO

Os sistemas biológicos resultam de um processo evolucionário. A sofisticação, robustez e adaptabilidade dos sistemas biológicos representam uma motivação para replicação dos mecanismos presentes para a geração de *hardware* e *software* [27]. A observação dá-nos uma resposta a alguns problemas da computação porque estes necessitam de ser bem definidos, geralmente previsíveis e resolúveis em tempo razoável. Os métodos normais por vezes não servem os propósitos e a bio computação surge como uma possível resposta. Podemos afirmar que a área de bio computação ou computação biomolecular iniciou-se há muito, desde que as primeiras formas de vida evoluíram. O que distinguiu estes simples organismos da simples coleção de moléculas inanimadas foi a capacidade de processar informação, e a sua capacidade de computação química que continua até os dias de hoje a servir de suporte à vida [28]. Uma área que junta e junta-se à bioquímica, matemática e computação, ou seja, uma junção da biologia e computação. Considera-se algumas subáreas como: computação baseada em moléculas biológicas em vez do silicone, bioinformática, modelação biológica e biologia

computacional. O processamento da informação recolhida é a chave para a inteligência de cada elemento [29].

5.2. INSETOS SOCIAIS

Vários insetos, como formigas e abelhas, têm um complexo sistema social. O seu comportamento em grupo é uma de várias propriedades emergentes das colónias dos chamados insetos sociais. Os insetos sociais diferenciam-se em estrutura, função e comportamento em castas sendo as maiores a reprodutiva (p. ex. a rainha) e os estéreis (operários e soldados). Além de desempenharem a função básica de reprodução, os membros da casta reprodutiva geralmente escolhem o local da nova colónia e escavam as primeiras galerias. Os operários cuidam dos ovos e larvas, colhem a comida para os outros membros e constroem/reparam o ninho, enquanto os soldados defendem a colónia dos predadores [30]. Em sentido geral qualquer inseto que interage com outros indivíduos da mesma espécie pode ser considerado como social. Mas, por regra, os entomologistas não consideram este comportamento como justificação suficiente para classificar uma espécie como verdadeiramente social (eusocial). Para qualificar-se como eusocial uma espécie tem de exibir todas as quatro seguintes características:

- Partilhar uma colónia comum;
- Cooperação dos elementos da espécie para cuidar dos mais novos;
- Divisão de trabalho reprodutiva – estéreis, ou menos fecundos, trabalham em benefício dos poucos que são reprodutores;
- Sobreposição de gerações.

Estes insetos sociais são melhor exemplificados pelas térmitas (isóptera) e formigas (formicídea) e várias abelhas e vespas (himenóptera) [31]. A observação destes comportamentos apresenta uma interessante abordagem à resolução de alguns problemas. Os SOAs (*Swarm-based Optimisation Algorithms*) apresentam semelhanças com estes métodos da natureza para encontrar uma solução ótima para diversos problemas.

5.3. ANT COLONY OPTIMIZATION

Foi proposto por Marco Dorigo [32] em que o seu algoritmo procurava encontrar um caminho ótimo num grafo. Este baseava-se no comportamento das formigas para encontrar o caminho entre a sua colónia e a fonte de alimento. É uma técnica probabilística para a resolução de problemas computacionais reduzíveis a pesquisa de grafos. Uma colónia de formigas é um sistema distribuído que, apesar da simplicidade, dos indivíduos apresentam uma grande estrutura social. Com esta premissa é possível conseguir desempenhar tarefas complexas que, em alguns casos, ultrapassam a capacidade de um indivíduo. A acuidade visual de várias espécies de formigas é rudimentar e algumas são mesmo cegas. Uma das grandes descobertas da interação das formigas é que a maior forma de comunicação entre formigas e entre estas e o ambiente é baseada na utilização de químicos produzidos pelas formigas. Estes são as feromonas. É uma forma distinta de comunicação de outras

espécies, como a humana, que se baseiam nos sentidos visuais e acústicos [33]. A trilha de feromonas é importante para a sociedade de algumas espécies de formigas, *Lasius niger* e *Iridomyrmex humilis* por exemplo, pois a sua especificidade implica a marcação de caminhos entre, por exemplo, comida e a colónia. Ao descobrir a trilha as formigas operárias podem descobrir o caminho para uma fonte de comida descoberta por outras formigas. Esta forma coletiva de comportamento de criar/seguir a trilha influenciada pela trilha química deixada por outras formigas é a base do ACO. Cada formiga tenta encontrar uma rota entre o formigueiro e a fonte de alimento. De [34] é a conclusão que em percursos com a mesma distância existe uma flutuação até a passagem das formigas atingir um estado estável, Figura 5.1. A técnica consiste na libertação de feromonas no percurso que a formiga percorre. No princípio a formiga vagueia à procura de alimento, deixando um rasto de feromonas. Quando encontra o alimento volta ao formigueiro novamente deixando um rasto de feromonas. Se encontrou feromonas no caminho irá seguir a mesma trilha. Ao chegar ao formigueiro volta a sair à procura de alimento. As feromonas evaporam-se ao longo do tempo, se não for a trilha reforçada por outras formigas, esta irá desaparecer.

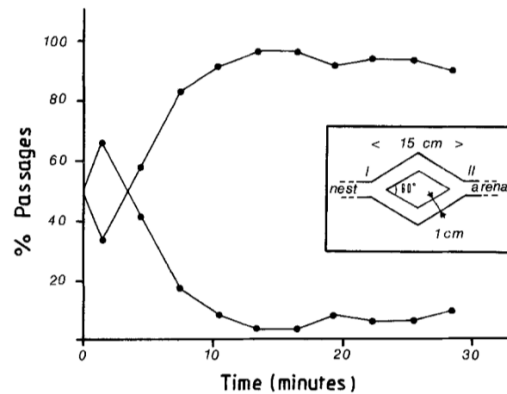


Figura 5.1 – Percentagem da passagem das formigas num dos percursos, de [34]

Essencialmente é a técnica de escolha do caminho mais curto. Recordamos que uma formiga isolada essencialmente move-se de forma aleatória, se detetar uma trilha de feromonas poderá segui-la reforçando-a com as suas próprias feromonas. Este processo cria um feedback positivo, Figura 5.2, que se reforça com o aumento de formigas a seguir a mesma trilha [35].

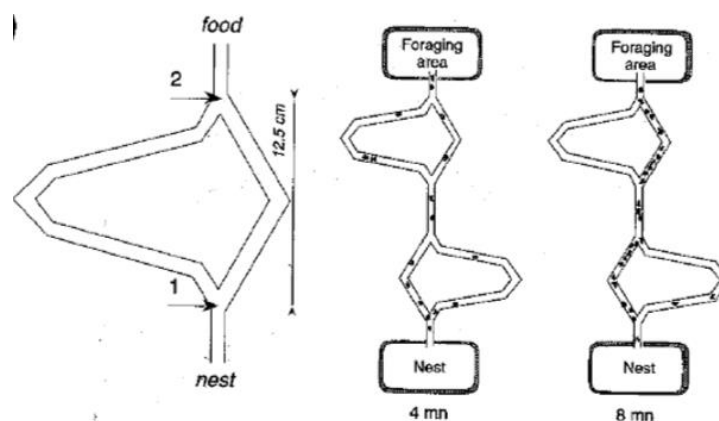


Figura 5.2 – Passagem das formigas num dos percursos [35]

5.4. BEE ALGORITHM

É um algoritmo baseado na população. Desenvolvido por Pham [36] em 2005 procura emular o comportamento recolector de uma colónia de abelhas. Esta pode englobar grandes distâncias em várias direções para explorar e encontrar alimento. A colónia irá prosperar se tiver acesso a boas fontes de alimento. Se os operários visitarem mais vezes estas fontes e menos outras a eficiência será melhor [37].

O processo de recolha inicia-se com a busca de locais promissores de flores. As abelhas, de forma, aleatória movem-se de um local para outro de forma a encontrar um local mais apropriado. Quando retornam à colmeia as abelhas operárias, que encontraram locais que apresentam determinadas características desejáveis (a combinação de vários constituintes como o conteúdo de açúcar), depositam o néctar ou pólen e iniciam uma dança conhecida como “*waggle dance*”. Esta é essencial para a comunicação na colónia pois contém três importantes dados – a direção, a distância e a qualidade – do local onde a abelha obteve o seu néctar. Esta informação permite que as operárias da colmeia se desloquem de forma precisa ao local de recolha [37]. O conhecimento de cada indivíduo da colmeia do ambiente exterior é recolhido da observação da dança. Esta permite que a colónia avalie o mérito de cada local a partir da qualidade e da energia necessária para a sua recolha [38]. Após cada dança a operária voa de novo para o seu local juntamente com eventuais novas operárias. Assim um maior número de operárias serão enviadas para os melhores locais fazendo que a colmeia colete com maior eficiência o melhor alimento. Nestas recolhas o nível de qualidade do local é monitorizado para quando do regresso a dança possa refletir eventuais alterações.

O algoritmo inicia com n abelhas colocadas aleatoriamente no espaço de procura. A qualidade de cada local é avaliada no passo seguinte. As abelhas com maior qualidade são selecionadas e os locais visitados por estas são escolhidos. Nos passos seguintes o algoritmo conduz pesquisa nos locais selecionados atribuindo mais abelhas aos locais com melhor qualidade. Esta atribuição pode ser feita com base na qualidade dos mesmos ou utilizada esta para determinar a probabilidade de atribuição. Na seleção das abelhas mais aptas (melhor qualidade dos locais) só a de melhor qualidade será selecionada para nova população. Esta restrição não existe na natureza, mas reduz o número de pontos de exploração. As restantes são atribuídas aleatoriamente no espaço de pesquisa para procurar novas possíveis soluções.

Estes passos serão repetidos até ser atingido um critério determinado. Após cada iteração a colónia irá ter duas partes na nova população – representantes de cada vizinhança selecionada e outras abelhas atribuídas a pesquisa aleatória [36]. A Figura 5.3 resume o processo.

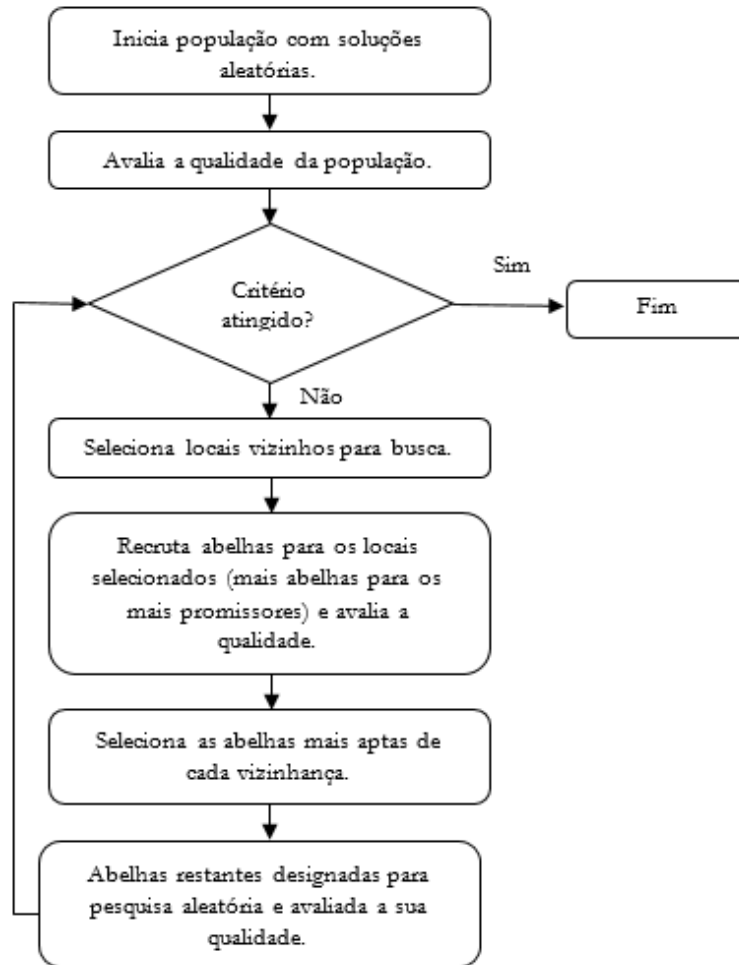


Figura 5.3 – Bee Algorithm

5.5. PSO

O algoritmo *Particle Swarm Optimization* foi introduzido por Eberhart e Kennedy em 1995 [39]. Neste uma determinada população representa uma possível solução. Cada partícula é movimentada por regra(s) simple(s) e não existe a garantia de uma solução satisfatória seja descoberta. A PSO tem raízes em duas metodologias principais. A mais óbvia relaciona-se com a vida artificial em geral e com o voo de pássaros, cardume de peixes e teoria de enxame em particular. A computação evolucionária e programação genéticas e evolucionária também tem influência no seu desenvolvimento [40]. O algoritmo foi desenvolvido após a observação de várias simulações desenvolvidas para interpretação do movimento dos organismos naturais. O conceito geral segue os seguintes elementos:

- Conjunto de partículas;
- Cada partícula ocupa uma posição no espaço de procura;
- A robustez da cada partícula representa qualidade da sua posição;
- As partículas percorrem o espaço de procura com uma certa velocidade;
- A velocidade (direção e deslocamento) é influenciada pela sua melhor posição conhecida e a melhor posição descoberta pelas partículas vizinhas;
- Repetir os passos até convergir numa posição ótima.

O processo é resumido na Figura 5.4.

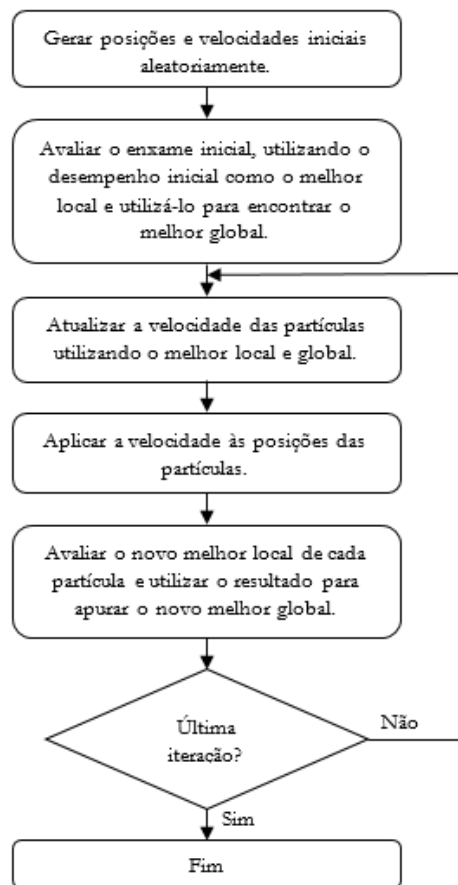


Figura 5.4 – Algoritmo PSO

5.6. MODELO SECI

Este modelo proposto por Ikujiro Nonaka e Hirotaka Takeuchi [41] explicita um processo de criação de conhecimento. Este pretende compreender a natureza da dinâmica de criação de conhecimento e a sua gestão. Do original *Socialization Externalization, Internalization e Combination*. Sendo assim a criação do conhecimento será um processo de contínua transferência, combinação e conversão de diferentes tipos de conhecimento enquanto os elementos de um grupo agem, interagem e aprendem. Esta criação é ajudada por dados e informação relevante que permite melhorar as decisões bem como providenciar base para criação de posterior novo conhecimento. O trabalho apresenta dois tipos de conhecimento:

- *Conhecimento explícito*: formal e codificado: todos os elementos têm acesso ao mesmo, é armazenado e poderá ser revisto, atualizado ou eliminado;
- *Conhecimento tácito*: o conhecimento intuitivo, baseado na experiência e muitas vezes depende do contexto.

O conhecimento explícito é muito aproximado à visão ocidental que vê uma determinada organização como uma máquina de processamento de informação. Esta visão promove o conhecimento útil como aquele que é sistemático e formalizado, ou seja, quantificável. A visão oriental (no caso presente

Japonesa) é a apropriação que a criação de novo conhecimento não é a penas o processar de informação objetiva. Depende, também, de explorar a informação tácita e subjetivas das opiniões, intuições e palpites dos indivíduos que compõem a empresa e de a fazer chegar a toda a empresa para que seja utilizada. A visão vai de encontro à assunção que uma empresa não é uma máquina, mas um organismo que, como um indivíduo, pode ter um senso coletivo de identidade e propósito. O conhecimento tácito é altamente pessoal e por consequência muito difícil de formalizar e logo de comunicar aos outros. Tomando as palavras do filósofo Michael Polanyi “podemos saber do que dizemos”. Este conhecimento também está intimamente ligado à ação e ao cometimento do indivíduo a um determinado contexto. Seja uma profissão, tecnologia ou atividades de um grupo ou equipa. Consiste parcialmente em capacidades técnicas – o tipo de habilidade informal difícil de descrever encapsulado no termo “*know how*”. Um artesão com anos de experiência desenvolve um manancial de capacidades. Mas é por vezes incapaz de articular os princípios técnicos ou científicos que apoiam todas as suas capacidades. Este conhecimento tem um importante aspeto cognitivo, pois consiste em modelos mentais, crenças e perspetivas tão profundamente enraizadas que são tomadas por garantidas sendo difícil de o articular. O modelo compreende quatro pontos:

- Socialização (*Socialization*): Tácito para tácito – o conhecimento é passado através de prática, acompanhamento, imitação e observação;
- Externalização (*Externalization*): Tácito para explícito – é considerado o mais complexo de atingir e por sua vez o mais importante mecanismo de conversão. O conhecimento é codificado em documentos, manuais, procedimentos, entre outros, para ser possível de distribuir pela organização. Como este é o mais complicado de codificar a completa viabilidade deste passo é debatida;
- Combinação (*Combination*): Explícito para Explícito – a forma mais simples, as fontes de conhecimento são combinadas para criar novo conhecimento;
- Internalização (*Internalization*): Explícito para tácito – as fontes de conhecimento são utilizadas e apreendidas modificando o conhecimento tácito dos elementos.

O conhecimento é continuamente convertido e criado com a prática e aprendizagem dos utilizadores. A aplicabilidade do modelo tem sido ligada à cultura tanto organizacional como nacional. A Figura 5.5 mostra este modelo. Quanto mais decorrer o processo melhor será o conhecimento criado.

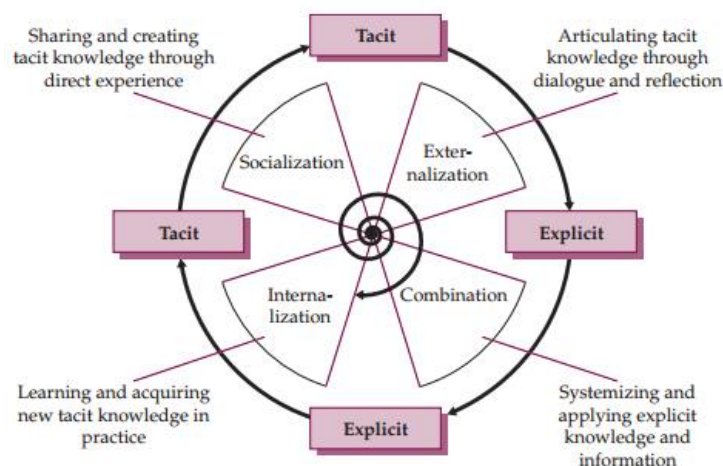


Figura 5.5 – Modelo SECI [42]

5.7. CONCLUSÃO

Neste capítulo foi discutida a capacidade um grupo de utilizar cooperação para atingir um objetivo. Essa capacidade foi observada na natureza e utilizada para construir algoritmos que realizam diversas funções em sistemas computacionais. Estes algoritmos apresentam soluções para problemas existentes utilizando técnicas presentes na natureza. Estas técnicas são utilizadas pelos diversos animais na sua vida social. Por último abordamos um procedimento uma visão que aproxima uma organização a esta técnicas tornando a organização uma comunidade que trabalha para um objetivo. Este conhecimento coletivo presente é importante pois a comunidade irá depender dele para o seu correto sustento. Esta importância é aparente no ACO pois a sobrevivência da colônia depende do conhecimento do melhor percurso para a fonte de alimento. Seguindo de várias definições de inteligência artificial o conhecimento é parte relevante do agente e nos casos representados no capítulo existe mais do que o conhecimento de cada agente. Existe um *conhecimento coletivo* criado pela interação dos agentes no seu mundo. É necessário a representação para que o agente possa decidir, ou seja raciocinar, para tomar as suas decisões. Assim é necessário estudar como este conhecimento poderá ser representado.

6. CONHECIMENTO COLETIVO

Como vemos dos exemplos anteriores um conjunto de indivíduos com pouco, ou nenhuma, inteligência consegue efetuar tarefas de alguma complexidade mesmo que individualmente não tenha essa consciência. As formigas encontram o melhor caminho até à fonte de alimento seguindo o rasto de feromonas deixado pelas companheiras que irá ser otimizado conforme o aumento de formigas seguindo o mesmo caminho. A comunicação é feita pela alteração do ambiente e o indivíduo não tem uma consciência do ambiente em termos gerais, apenas que deve seguir o melhor caminho. Este foi otimizado pelas passagens anteriores do mesmo e dos outros indivíduos. As abelhas comunicam de uma forma um pouco diferente. Comunicam a sua descoberta para que todos na colmeia a possam ver e tenta recrutar mais indivíduos para que sigam no caminho indicado. Esta comunicação é feita com recurso a uma dança com propriedades muito específicas em que o indivíduo que a executa tem de ter um conjunto de ações que tentam influenciar a assistência, como dar a indicação do caminho a percorrer. Assim sendo este indivíduo tem uma consciência do mundo em geral pois precisa de dar indicações específicas de como encontrar a fonte de alimento. Uma descrição genérica de conhecimento dada por [20] indica o conhecimento como informação armazenada ou modelos utilizados por uma pessoa ou máquina para interpretar, prever e responder apropriadamente ao mundo exterior. Assim a representação do conhecimento apresenta duas características: a informação explícita e como a informação é codificada para posterior utilização. Então a representação de conhecimento tem um objetivo: fornecer uma adequada representação do conhecimento. Esta representação irá afetar a solução adequada ao problema [19].

6.1. REPRESENTAÇÃO DO CONHECIMENTO E RACIOCÍNIO

A formalização do conhecimento declarativo foi desenvolvida, pelos matemáticos e lógicos, muito antes do aparecimento dos sistemas de computação. Mas estes preocupavam-se com a formalização da matemática [43]. A importância do conhecimento não matemático, na IA, não era enquadrada na perspectiva da representação do conhecimento pois a sua abrangência não era suportada pela falta expressividade da linguagem formal. Por outro lado, a maioria dos lógicos não se preocupavam com o raciocínio automático, mas apesar de tudo a matemática lógica clássica exerceu grande influência no campo da representação do conhecimento. A representação do conhecimento pode ser encarada como uma subárea da inteligência artificial que se ocupa com a compreensão, desenho e implementação de forma de representação da informação em sistemas computacionais para que estes, programas ou agentes, possam utilizar esta informação:

- Para derivar informação implícita por esta;
- Para comunicar com humanos através de linguagem natural;
- Para decidir os que fazer de seguida;
- Para planear tarefas seguintes;
- Para resolver problemas que, normalmente, requerem as competências de um ser humano.

O raciocínio implica derivar informação a partir de informação já existente. Assim a representação do conhecimento não será completa sem a capacidade de raciocinar sobre este.

6.2. ABORDAGEM CONEXIONISTA

Baseia-se na hipótese de causa-efeito, segundo a qual um modelo suficientemente preciso do cérebro humano é suficiente para reproduzir a inteligência que um humano possui. Essa abordagem trata de problemas imprecisos, mas que podem ser definidos através de exemplos (como reconhecimento de caligrafia), e sua principal contribuição são as redes neuronais. Estas são modelos inspirados no cérebro humano que possuem capacidade de aquisição, manutenção e utilização do conhecimento. Estes modelos, computacionais, distribuídos são compostos por um conjunto de unidades de processamento, os neurónios, dispostas em camadas interligadas por diversas conexões, as sinapses. Às conexões são associados pesos que armazenam o conhecimento representado no modelo e servem para ponderar e avaliar a entrada recebida por cada neurónio da rede.

O processo de resolução de problemas passa por um período de aprendizagem seguido de uma fase de teste. Na fase de aprendizagem existem várias abordagens:

- **Aprendizagem supervisionada** - existe um sistema que indica explicitamente se o comportamento da rede é bom ou não;
- **Aprendizagem não supervisionada** - existe um sistema verifica o comportamento da rede para casos semelhantes.

As principais características do modelo passam pela aprendizagem por exemplos (padrões de treino), capacidade de adaptação e aprendizagem, generalização, agrupamento de dados, tolerância a falhas e auto-organização.

6.3. ABORDAGEM EVOLUCIONÁRIA

Baseia-se na teoria evolutiva de Darwin, a hipótese é que podemos modelar sistemas inteligentes simulando a evolução de uma população de indivíduos aleatórios, que transportam genes com informação suficiente para dar origem à solução de um problema, usando operações genéticas de recombinação e mutação. Essa abordagem trata de problemas de otimização (como escalonamento de produção) e sua principal contribuição são os algoritmos genéticos. Este é um algoritmo estocástico que procura simular o processo evolução biológica natural seguindo o princípio de Darwin da sobrevivência do mais adaptado e seleção natural. Os problemas são definidos com como cromossomas representados, usualmente, por *strings* binárias com as principais operações *crossover* e mutação. Este ciclo é representado por três estádios sequenciais:

- Geração da população (estados representados pelos cromossomas)
- Evolução genética através do *crossover* seguido de mutação;
- Seleção do melhor candidato gerado pela população.

No passo inicial são identificados alguns casos iniciais, no passo seguinte evolui novos cromossomas para no final os melhores candidatos serem escolhidos. De seguida os passos são repetidos um número de vezes finito a fim de obter a solução para o problema proposto. A Figura 6.1 mostra a troca de informação entre os genes.

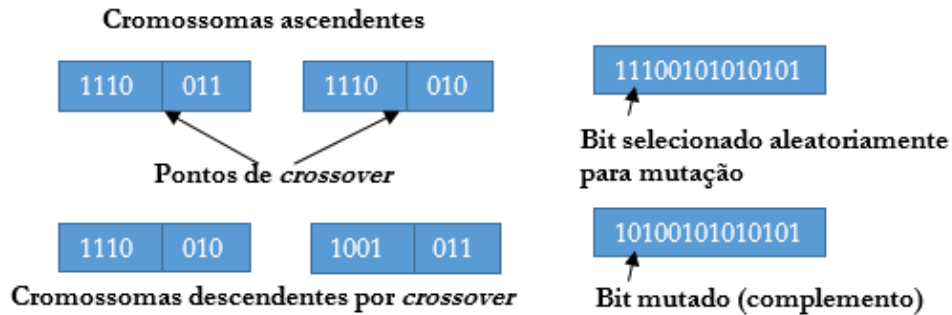


Figura 6.1 – Troca de informação genética por *crossover* e mutação

Esta abordagem encontra muitas aplicações nos campos de pesquisa inteligente, aprendizagem automática e otimização [44].

6.4. ABORDAGEM SIMBÓLICA

Baseia-se na hipótese do sistema de símbolos físicos, segundo a qual um conjunto de estruturas simbólicas e um conjunto de regras de manipulação dessas estruturas são os meios necessários e suficientes para se criar inteligência. Essa abordagem trata problemas bem definidos (como planeamento de tarefas) e sua principal contribuição são os sistemas especialistas. Para esta abordagem é necessário identificar o conhecimento do domínio (modelo do problema). Representá-lo utilizando uma linguagem formal de representação e implementar um mecanismo de inferência para a utilização desse conhecimento. O agente racional age num ambiente segundo certos princípios, precisa de ter conhecimento sobre as propriedades relevantes do mundo, como este evolui, como identificar os estados desejáveis, como medir o sucesso das suas ações e como avaliar o seu próprio conhecimento. O resultado da aprendizagem é um conjunto de afirmações/crenças/processos. Os inconvenientes é que é muito orientado ao tratamento de informação discreta e existem dificuldades em problemas não lineares.

A dificuldade centra-se em representar o conhecimento para que possa ser manipulado tanto para utilizá-lo como para a sua atualização. Quando falamos de conhecimento apresenta-se, primeiro, o que é o conhecimento em si. Um problema que tem vindo a ocupar a humanidade desde há muito. Sendo que se pretende representar o conhecimento coletivo de uma comunidade este problema aumenta. Dessa forma é necessário representar o conhecimento através de símbolos:

- Vantagens: poder discursar sobre o conhecimento através de símbolos não modificando o original;
- Desvantagens: os símbolos têm de ter significado no domínio.

Para começar é necessário dar um significado aos símbolos. O que não é uma tarefa simples pois dizer que algo significa algo leva-nos, primeiro, a questionar o que se entende por significa. De forma intuitiva o significado de algo será encontrar o conceito ou noção do mesmo. Claro que agora poderemos entrar em ciclo procurando o que se entende por noção ou conceito de algo. Esse conceito ou noção pode ser representado por um símbolo que apresente o significado de forma perceptível por todos.

Para utilizar este paradigma é necessário enquadrar três pontos importantes:

- Identificar o conhecimento do domínio;
- Fazer a sua representação através de uma linguagem formal de representação;
- Implementar um mecanismo de inferência para utilizar esse conhecimento.

No sistema um agente racional irá agir num ambiente segundo determinados princípios racionais para atingir determinados objetivos. Assim terá de ter conhecimento sobre:

- Propriedades relevantes do ambiente em que se situa;
- Como este ambiente evolui;
- Quais os estados desse ambiente;
- Quais as consequências das suas ações;
- Como medir o impacto dessas ações;
- Como avaliar o seu próprio conhecimento.

6.5. CONCLUSÃO

Neste capítulo abordamos o conhecimento coletivo como parte integrante de uma comunidade de agentes. No capítulo anterior afirmamos a emergência de conhecimento através da interação dos elementos de uma comunidade e a necessidade de proceder à sua representação. Dessa forma neste capítulo, também, abordamos várias formas da sua representação nos paradigmas da inteligência artificial. As diversas representações apresentam características que as fazem apropriadas a diversas utilizações. A abordagem simbólica apresenta vantagens relevantes, como explicar o porquê do processo e do resultado, mas tem um maior peso na formalização de toda a plataforma que sustenta a representação e utilização. Este peso representa dificuldades de implementação e de espaço temporal para a representação e resposta respetivamente.

Parte III - Bases Paradigma Simbólico

Apresentação dos conceitos base. Representação e Raciocínio. Revisão de Crenças e Argumentação. Dinâmica do Conhecimento e Teoria da Decisão. Apresentação do Modelo e Conclusões.

7. REPRESENTAÇÃO DO CONHECIMENTO

Como já abordado em capítulos anteriores a IA aborda o conhecimento e ação efetuada sobre e do mesmo. Neste capítulo abordaremos os conceitos relacionados com a representação do conhecimento. Uma das definições da IA descreve como o estudo do comportamento inteligente através de meios computacionais. Dessa forma a Representação e Raciocínio é a parte que se ocupa de como um agente utiliza o que *sabe* na *decisão* de como agir. Assim iremos descrever algumas das estruturas simbólicas utilizadas para suportar a representação do conhecimento bem como com os processos computacionais para raciocinar através dessas representações.

7.1. CONHECIMENTO: REPRESENTAÇÃO E RACIOCÍNIO

Em [45] definem-se três conceitos chaves: *Conhecimento*, *Representação* e *Raciocínio*. Iremos discutir estes três conceitos chaves para a IA.

7.1.1. CONHECIMENTO

O conhecimento é um conceito complexo de abordar pois poderá ser tratado de várias formas: filosófico, psicológico e computacional. McCarthy, em 1959, afirma que um agente para um sistema que utilize o conhecimento trabalha com “*aquilo que já sabe/conhece*”. Ora isso será o conhecimento e terá de ter uma estrutura capaz de representar o conhecimento. De seguida resumimos, a partir [45], uma visão sobre o que é o conhecimento. O que é o conhecimento é discutido há muito, desde os filósofos Gregos, e continua a ser um tópico não totalmente apreendido. Quando afirmarmos “*Eu sei ...*” fazemos uma declaração sobre uma propriedade consideramos saber. Ou seja, criamos uma relação entre uma entidade e uma proposição que é a ideia expressa pela afirmação. Essa relação também pode ser considerada como conhecimento. O problema, se assim se pode afirmar, do conhecimento é a natureza das proposições. O que se apreende dessas afirmações é que são entidades abstratas que podem ser verdadeiras ou falsas, corretas ou incorretas. Esta proposição representa um determinado julgamento feito por alguém e que entende o mundo de uma forma e não de outra. A proposição é reflexo desse entendimento. A relação entre a proposição e o sujeito pode ter contornos ligeiramente diferentes. Tendo em atenção à linguagem utilizada o sujeito pode “*saber*”, “*esperar*”, “*confiar*”, “*desejar*”, “*recear*” entre muitas outras expressões linguísticas, o que indica uma variação na relação entre ele e a proposição. Isto não invalida que a proposição denota uma determinada visão do mundo, verdadeira, do ponto de vista do sujeito. Estas denotam-se por *atitudes proposicionais*.

Outras frases não apresentam estas verdades através de proposições explícitas. Ao dizer que “*Eu sei como chegar a...*” ou que certa pessoa tem uma determinada habilidade ou que conhece bem uma outra pessoa declaramos um tipo de conhecimento sem utilizarmos proposições de forma clara pois apenas declaramos um conhecimento, possivelmente, indireto. Uma outra noção importante é a *crença*. A afirmação “*Eu acredito que...*” representa uma relação com “*Eu sei que...*”. A primeira reflete a

realidade do sujeito quando o seu conhecimento pode não ser necessariamente correto ou suportado. Este apresenta a realidade de que podemos atribuir um conjunto de atitudes proposicionais que diferem apenas no nível de convicção atribuído a cada uma destas proposições. Ponto assente é que qualquer uma destas proposições apresenta a visão que o sujeito tem do mundo à sua volta. Este considera o mundo de uma forma e não de outra.

7.1.2. REPRESENTAÇÃO

Após discutirmos o conhecimento necessitamos de uma estrutura capaz de suportar o conhecimento para que possa ser acedido e/ou manipulado. McCarthy, em 1959, afirma que um agente “*tem de ser capaz de ser informado*”, ou seja, tem de ser capaz de incorporar novo conhecimento naquele que já tem. Para isso a estrutura de representação não pode, simplesmente, suportar o conhecimento estático. Tem, também, de suportar funções de manipulação. A partir de [45] tiramos que a representação é um outro tópico filosoficamente excitante. De uma forma simples a representação é a relação entre elementos de domínios distintos em que o segundo toma o lugar do primeiro, ou seja, representa o primeiro de uma outra forma. Um dos domínios é mais concreto e imediato e/ou acessível que outro. Podemos apresentar um desenho ou diagrama que apresente um conceito ou eleger alguém para uma assembleia que fale por um conjunto de pessoas. O elemento de maior interesse, para nós, é o *símbolo*. Este é um carácter ou conjunto de caracteres de um determinado alfabeto. Ao utilizarmos, por exemplo, um número este representa o mesmo conceito mesmo considerando vários alfabetos ou contextos. Como se espera esta representação é mais simples de trabalhar do que com o que os símbolos representam.

Os elementos mais abstratos são os mais complicados de representar através de símbolos. No presente trabalho interessamos-nos especialmente pelos símbolos que representam as proposições. Pois apesar das frases serem bem definidas sintática e semanticamente a proposição pode ser abstrata. Dessa forma a representação de conhecimento é o campo de estudo ocupado com a utilização de símbolos formais para representar a coleção de proposições que um determinado agente mantém, o seu conhecimento. Uma vertente a considerar é a existência de um conjunto de proposições que não são representadas. O papel do raciocínio será tentar superar esta divisão entre o que se acredita e aquilo que é efetivamente representado.

7.1.3. RACIOCÍNIO

Novamente socorremo-nos McCarthy que afirma o agente “*tem de ser capaz de deduzir por si mesmo um conjunto de consequência imediatas*”. O agente terá de ser capaz, a partir do conhecimento que detém, ser capaz de chegar a conclusões não presentes. Em [45] o processo de raciocínio, em geral, é a manipulação de símbolos que representam um conjunto de proposições para produzir representações de novas proposições. Aqui entra o facto de que os símbolos são mais acessíveis que as proposições que representam. Terão de ser concretos para que se possam manipular e para que se possam contruir

representações de novas proposições. Gottfried Leibniz, no séc. XVII, propôs o raciocínio com uma forma de cálculo, como a aritmética, mas através de símbolos. Esta visão vem do facto de se poder retirar conclusões através de alguma manipulação das proposições iniciais. Este raciocínio é chamado de *lógica de inferência* porque a proposição final representa a conclusão final das proposições iniciais.

7.1.4. REPRESENTAÇÃO E RACIOCÍNIO

A representação e raciocínio são dois processos intimamente interligados e dependem um do outro. Para efetuar um raciocínio é necessário que se represente o conhecimento para ter sobre o que raciocinar. Apenas a representação é limitada pois sem raciocínio não é possível tomar decisões ou modificar a conhecimento em resposta às mudanças que possam ocorrer. Em [45] encontramos diversas noções sobre os dois processos. O conhecimento é relevante nos sistemas de IA pois, geralmente, é utilizado para descrever o comportamento de sistemas suficientemente complexos através de vocabulário como *crenças, desejos, objetivos, intenções* e outros. Podemos observar estes processos na forma como observamos o desenrolar de um jogo. Observamos uma jogada e convencemo-nos que o programa efetuou uma determinada ação porque *pensou* que seria a melhor. Ou seja, ao nível do que o filósofo Daniel Dennett retratou de *posicionamento intencional*, compreender o programa através dos seus objetivos imediatos relativos às suas crenças, e planos/intenções a longo prazo. Porque é mais simples do que descrever o processo formal que é subjacente ao funcionamento do programa. Esta visão pode ser uma pouco limitadora pois poderá levar a considerarmos que estamos perante algo mais complexo do que é na realidade. Bastará lembrarmo-nos de um termóstato que *sabe* que o quarto está frio e *quer* aquecê-lo. A antropomorfização dos sistemas é, geralmente, inapropriada pois existe explicação mecânica/elétrica simples do funcionamento do sistema. Mas tudo isto não apresenta o que é a representação do conhecimento no sistema. Ao observarmos o posicionamento intencional não nos mostra o que é ou não representado simbolicamente no sistema. Pensando sobre o jogo de xadrez podemos representar a posição das peças simbolicamente, mas, por exemplo, o objetivo de movimentar o Cavalo pode apresentar diversas dificuldades. Este objetivo poderá ser realizável apenas após um complexo jogo de interações das propriedades e aspetos do programa, bibliotecas de movimentos e jogos, funções de avaliação entre outros. Contudo consideramos que o sistema *tem* este objetivo se este é explicado convenientemente pelo comportamento do programa.

Então o papel da representação simbólica será permitir a construção de um sistema em que esta representação contém duas importantes propriedades. A primeira é que, visto do exterior, podemos entender as suas intenções por proposições. A segunda, que o sistema seja desenhado para se comportar de uma forma por causa dessas representações simbólicas. Estas propriedades abarcam o que o filósofo Brian Smith designou de *Hipótese de Representação de Conhecimento*. Este implica que devemos construir um sistema em que o posicionamento intencional é baseado na representação simbólica. Estes sistemas são chamados de *Sistemas Baseados em Conhecimento* e as representações

simbólicas de *Bases de Conhecimento – Knowledge Bases – KB*. Grande parte da IA envolve a construção deste tipo de sistemas. Sistemas que em que as conclusões derivam, em parte, do raciocínio sobre conhecimento representado explicitamente. Esta representação é feita através duma coleção de estruturas simbólicas que representam o que o sistema acredita e raciocina durante a operação do sistema.

A representação tem a finalidade de fornecer aos sistemas de conhecimento a robustez de um sistema aberto. Seria mais simples incorporar o conhecimento diretamente no programa sem este ter de passar pelo raciocínio sobre a sua *KB*. Isto porque estes sistemas tendem a ser um pouco mais lentos comparados com os tradicionais programas procedimentais. Podemos observar que realizamos tarefas complexas sem pensar sobre o que fazemos, mas, fazendo-o diretamente; como, por exemplo andar de bicicleta. Não pensamos em todos os pormenores, mas simplesmente utilizamos a bicicleta para nos deslocar. Se tivéssemos que pensar em todos os pormenores provavelmente nem nos mexíamos. Mas temos uma tarefa definida, andar de bicicleta, que não é um processo em aberto, mas definido. Depois de aprendermos uma determinada tarefa deixamos de pensar nos pequenos passos necessários e fazemos, como se habitualmente se diz, sem pensar. Podemos dizer, por exemplo, que entre dois jogadores de xadrez o praticante iniciante parece pensar sobre tudo para agir enquanto o veterano parece simplesmente ver o que é necessário e agir, isto porque o veterano já jogou e observou muitos jogos. Isto ilustra o processo da representação pois quando existe um processo em que o processo para a conclusão é em aberto, ou seja, não existe uma solução única ou definida nem todas as possibilidades especificadas é preferível ter um conhecimento sobre o qual se possa raciocinar para chegar a uma conclusão mais correta.

Um sistema de conhecimento mais facilmente recebe nova informação pois será mais uma cláusula a inserir do que um sistema procedimental, pois não sabemos como esta irá ser distribuído pelo programa. A grande vantagem deste sistema é que, por construção, tem a possibilidade de receber informação e ajustar o seu comportamento de forma a incorporar a nova informação. Zenon Pylyshyn designou por *penetrabilidade cognitiva* às ações tomadas em consequência daquilo que acreditamos. A nossa resposta a determinados eventos é diferente conforme a conjectura efetuada sobre os mesmos. Tomemos como exemplo a reação a um alarme de incêndio. A primeira ação seria a de evacuar o local, mas poderíamos, por alguma razão, pensar que se tratava de um teste ou de mau funcionamento e não prestar atenção ao mesmo. A reação ao alarme é cognitiva pois está condicionada ao que acreditamos. Existem ações, por reflexo, que mesmo que acreditamos em algo vamos sempre agir mesmo que saibamos ser desnecessário.

A motivação de um sistema de conhecimento é conseguir que este dê uma resposta baseado naquilo que acredita em oposição àquilo que tem representado explicitamente. O que se introduz na *KB* é um conjunto de cláusulas que depois são aplicadas em variadas situações pelo sistema que incorpora a base de conhecimento.

O raciocínio permite que se retirem conclusões além dos dados que se podem obter da *KB*. Este processo seria como, apenas, uma consulta a uma base de dados. É necessário que se tirem conclusões não presentes explicitamente, mas, também, implicitamente. Este conceito, *relações de consequências*, advém que se as proposições representadas por um conjunto S suportam a proposição representada pela frase p quando a verdade de p é implícita pela verdade das frases em S . Para a noção de *consequência lógica* requer-se uma linguagem com o conceito de verdade ou mentira para que se aplique às frases dessa mesma linguagem. Porque saber P é acreditar que P é verdade.

Assim podemos assumir que as crenças, de um sistema de conhecimento, exibem todas, e apenas, consequências lógicas relativas ao explicitamente representado. Deste modo a função do raciocínio é de computar todas as consequências lógicas da *KB*. Esta é uma simplificação pois existem várias boas razões para não calcular as consequências. Poderá ser muito complicado computacionalmente decidir quais as consequências lógicas da *KB* em uso. As respostas de um sistema em tempo útil podem não ter todas as consequências, *logicamente incompleta*, ou consequências incorretas, *logicamente infundada*. Existem razões para que estes dois pontos sejam utilizados. Pode existir um p que não é consequência lógica da *KB*, mas pelo representado é aceitável assumir como verdade. Ou uma *KB* com informação de várias fontes que todas juntas não poderão ser verdade simultaneamente. Dessa forma, apesar das conclusões serem logicamente incompletas, o raciocínio será útil até que as contradições sejam, ou não, tratadas. Apesar destes pontos é vantajoso considerar o raciocínio desta forma mesmo que se pretenda divergir para um sistema mais robusto. Assim começamos por considerar os sistemas baseados em conhecimento como lógica e completamente fechado relativamente a inferências mesmo que não seja o mais correto [46].

7.1.5. O PAPEL DA LÓGICA

A lógica é relevante para a representação do conhecimento pois, de acordo com uma visão, a lógica é o estudo das relações de consequências – linguagens, condições de verdade e regras de inferência. São utilizadas ferramentas e técnicas da lógica simbólica. Especificamente a linguagem conhecida por *Lógica de Primeira Ordem – First-Order Logic – FOL*. Esta foi inventada pelo filósofo Gottlob Frege no início do séc. XX para a formalização de inferência matemática, mas tem sido utilizada na IA com propósito de representação de conhecimento.

A FOL não dá todas as respostas à representação do conhecimento, mas apresenta uma base que permite trabalhar com a mesma. Onde sobressai, na perspectiva da representação do conhecimento, é no que, Allen Newell, chamou de *nível de conhecimento*. A ideia é que um sistema baseado no conhecimento pode ser visto de dois níveis. O nível de conhecimento onde estão as informações sobre a linguagem de representação e sua semântica. No nível simbólico está a informação sobre os aspetos computacionais. No nível de conhecimento lida-se com os aspetos da adequada expressão de uma linguagem de representação e as características das suas relações de consequência, incluindo

a sua complexidade computacional. No nível simbólico, por sua vez, temos os aspetos da arquitetura computacional e as propriedades das estruturas de dados incluindo a sua complexidade algorítmica. As ferramentas da lógica simbólica formal parecem ser adequadas para a análise do nível de conhecimento de um sistema baseado em conhecimento [46].

7.2. LINGUAGEM DA LÓGICA DE PRIMEIRA ORDEM

Neste capítulo fazemos um resumo de [45] sobre a FOL pois antes de iniciar qualquer processo é necessário formalizar os conceitos envolvidos.

7.2.1. LINGUAGEM

Existem três elementos a considerar numa linguagem. A *Sintaxe* permite especificar que grupos de elementos, e em que ordem, são considerados como bem formados. Na representação do conhecimento é necessário ser extremamente claro sobre o que são *frases bem formadas* da linguagem pois são estas que expressam as proposições. Na *Semântica* explicita-se o que as expressões bem formadas significam. As expressões da linguagem, além de bem formadas, necessitam de ter um determinado significado. Pois estas têm de mostrar uma ideia clara sobre o mundo a ser expresso. A *Pragmática* aparece para especificar o nível de significado em que cada expressão é utilizada na linguagem. Na representação do conhecimento utilizamos as expressões mais significativas como parte da base de conhecimento de onde serão retiradas as inferências. Estes três aspetos aplicam-se, principalmente, a linguagens declarativas como as que são utilizadas para representação do conhecimento.

7.2.1.1. SINTAXE

Na linguagem existem dois tipos de símbolos: *lógicos* e *não lógicos*. Os símbolos lógicos são aqueles que têm um significado fixo na linguagem. Existem três destes símbolos lógicos:

1. *Pontuação*: (,) e ;
2. *Conectores*: \neg , \wedge , \vee , \exists , \forall e $=$;
 - A interpretação destes símbolos é a habitual, negação, conjunção, disjunção, existe, para todos e igualdade, respetivamente. Os símbolos \exists , \forall são chamados de quantificadores.
3. *Variáveis*: um conjunto de símbolos notados por x , y e z com sub e sobrescritos.

Os símbolos não lógicos são aqueles que têm um significado dependente da aplicação ou utilização:

1. *Símbolos de função*: um conjunto quase infinito de símbolos notados por minúsculas mistas e, mais frequentemente, por a , b , c , f , g e h com sub e sobrescritos;
2. *Símbolos de predicado*: um conjunto de símbolos notados por capitulação mista e, mais frequentemente, por P , Q e R com sub e sobrescritos.

Uma particularidade dos símbolos não lógicos é que cada um assume uma *aridade*, isto é, cada um contém um número, não negativo, que indica o número de *argumentos* que recebe. Este é utilizado na sintaxe da linguagem. Por convenção a , b e c são utilizados para funções de aridade 0 – denominados de constantes – enquanto g e h são utilizados para funções de aridade não zero. Os predicados das

funções de aridade zero são, por vezes, chamados de *símbolos proposicionais*. Existem dois tipos de expressões sintáticas válidas: *termos* e *fórmulas*. O primeiro é, geralmente, utilizado para se referir a algo no mundo e a fórmula para as proposições. O conjunto de termos da linguagem é o mínimo conjunto que satisfazem as seguintes condições:

- Toda a variável é um termo;
- Se t_1, \dots, t_n são termos e se f é um símbolo de função de aridade n então $f(t_1, \dots, t_n)$ é um termo.

O conjunto de fórmulas de linguagem é o conjunto mínimo que satisfaz as seguintes restrições:

- Se t_1, \dots, t_n são termos e P é o símbolo de predicado de aridade n então $P(t_1, \dots, t_n)$ é uma fórmula;
- Se t_1 e t_2 são termos então $t_1 = t_2$ é uma fórmula;
- Se α e β são fórmulas e x é uma variável então $\neg\alpha$, $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $\forall x.\alpha$ e $\exists x.\alpha$ são fórmulas.

As fórmulas dos dois primeiros tipos (sem conter outras fórmulas simples) são chamadas de *fórmulas atômicas* ou *átomos*. Uma variável pode ser restringida a uma fórmula se está dentro do âmbito de um quantificador ou livre caso contrário. Se x é uma variável, t é um termo e α uma fórmula a notação α_t^x indica a substituição de todas as ocorrências livres de x em α por t na fórmula. Se \vec{x} é uma sequência de variáveis, \vec{c} é uma sequência de constantes da mesma dimensão e α é uma fórmula cujas variáveis livres estão entre \vec{x} então $\alpha[\vec{x}]$ significa α mesmo e $\alpha[\vec{c}]$ significa α com cada variável x_i livre substituída pela correspondente c_i . Finalmente uma frase na linguagem é uma fórmula sem variáveis livres. As frases são utilizadas para representar conhecimento e o resto suporta a sintática.

7.2.1.2. SEMÂNTICA

Esta trata de explicar o que as expressões da linguagem significam. Envolve a especificação da declaração que uma frase da linguagem lógica faz sobre o mundo, de forma a se compreender o que a sua assunção significa. Esta apresenta um problema pois não é possível, realisticamente, especificar definitivamente o que cada frase da linguagem significa. Pois os símbolos não lógicos são dependentes do contexto em que são aplicados. Isto significa que o significado da frase depende da interpretação dos símbolos não lógicos envolvidos. O que pode ser muito complexo e difícil de precisar. Sendo assim torna-se necessário especificar o significado de uma frase como uma *função da interpretação do predicado e símbolo de função*.

Para tal especificação a visão do mundo poderia ser da seguinte forma:

- Existem objetos no mundo;
- Para qualquer predicado P de aridade 1, alguns dos objetos irão satisfazer P . Uma interpretação de P decide a questão decidindo para cada objeto se tem ou não a propriedade envolvida. Predicados de outra aridade são tratados de forma semelhante;
- Nenhum outro aspeto do mundo tem interesse.

Assim os significados dos símbolos não lógicos são abrangidos por estes três pontos e, por consequência, o significado das frases. É de notar que a linguagem não define os símbolos como encontraríamos num dicionário, mas, a partir da interpretação dada no contexto. Outras linguagens tratarão de outra forma os significados.

7.2.1.2.1. INTERPRETAÇÃO

Os significados são, geralmente, capturados através de interpretações específicas. Uma interpretação \mathfrak{I} , na linguagem, é um par $\langle \mathfrak{D}, \mathfrak{L} \rangle$ onde \mathfrak{D} é um conjunto, não vazio, de objetos, chamado de *domínio* da interpretação e \mathfrak{L} é um mapeamento, chamado *mapeamento de interpretação*, dos símbolos não lógicos para funções e relações sobre \mathfrak{D} .

O mapeamento vai atribuir significado aos símbolos do predicado da seguinte forma:

- para cada símbolo do predicado P de aridade n , $\mathfrak{L}[P]$ é uma relação n -ária sobre \mathfrak{D} , isto é, $\mathfrak{L}[P] \subseteq \underbrace{\mathfrak{D} \times \dots \times \mathfrak{D}}_n$.

O mapeamento vai atribuir significado aos símbolos de função da seguinte forma:

- para cada símbolo de função f de aridade n , $\mathfrak{L}[f]$ é uma função n -ária sobre \mathfrak{D} , isto é, $\mathfrak{L}[f] \in \underbrace{\mathfrak{D} \times \dots \times \mathfrak{D}}_n \rightarrow \mathfrak{D}$.

Por vezes é útil encarar a interpretação de predicados nos termos das suas funções características. Neste caso quando P é um predicado de aridade n e $\mathfrak{L}[P]$ é uma função n -ária para $\{0,1\}$ vem $\mathfrak{L}[P] \in [\mathfrak{D} \times \dots \times \mathfrak{D} \rightarrow \{0,1\}]$. A relação entre as duas especificações é que o tuplo de objetos é considerado numa relação sobre \mathfrak{D} se e só se a função característica sobre aqueles objetos tem valor 1. Também nos permite observar mais claramente como os predicados de aridade 0 (símbolos proposicionais) são tratados. Neste caso $\mathfrak{L}[P]$ será 0 ou 1. Podemos ver o primeiro caso como significando *falso* e o segundo *verdade*. Para o subconjunto proposicional da linguagem podemos ignorar \mathfrak{D} completamente e considerar um mapeamento, \mathfrak{L} , dos símbolos proposicionais para tanto 0 ou 1.

7.2.1.2.2. DENOTAÇÃO

Dada uma interpretação $\mathfrak{I} = \langle \mathfrak{D}, \mathfrak{L} \rangle$ podemos especificar quais os elementos de \mathfrak{D} que são denotados por qualquer termo de variável livre da linguagem. Para encontrar um objeto denotado por um termo em \mathfrak{I} utilizamos \mathfrak{L} para ter a função denotada e, finalmente, aplicamos essa função ao elemento de \mathfrak{D} produzindo um outro elemento de \mathfrak{D} . Para lidar com termos que incluem variáveis necessitamos de uma *atribuição de variável* sobre \mathfrak{D} , isto é, um mapeamento das variáveis da linguagem aos elementos de \mathfrak{D} . Então se μ é a variável de atribuição e x é a variável, $\mu[x]$ será um elemento do domínio.

Formalmente dada uma interpretação \mathfrak{I} e uma variável de atribuição μ a denotação do termo t , escrito por $\|t\|_{\mathfrak{I},\mu}$ é definido pelas seguintes regras:

- Se x é uma variável então $\|x\|_{\mathfrak{I},\mu} = \mu[x]$;
- Se t_1, \dots, t_n são termos e f uma função de símbolo de aridade n então $\|f(t_1, \dots, t_n)\|_{\mathfrak{I},\mu} = \mathfrak{F}(d_1, \dots, d_n)$ onde $\mathfrak{F} = \mathfrak{L}[f]$ e $d_i = \|t_i\|_{\mathfrak{I},\mu}$.

Observando as regras recursivas $\|t\|_{\mathfrak{I},\mu}$ é sempre um elemento de \mathfrak{D} .

7.2.1.2.3. SATISFAÇÃO E MODELOS

Dada uma interpretação $\mathfrak{I} = \langle \mathfrak{D}, \mathfrak{L} \rangle$ e a relação $\|\cdot\|_{\mathfrak{I},\mu}$ definida anteriormente podemos, agora, especificar que frases da linguagem são verdadeiras e quais são as falsas segundo esta interpretação. Formalmente dada uma interpretação \mathfrak{I} e a variável de atribuição μ dizemos que a fórmula α é satisfeita em \mathfrak{I} , escrita $\mathfrak{I}, \mu \models \alpha$ de acordo com as seguintes regras. Assumindo que t_1, \dots, t_n são termos, P é o predicado de aridade n , α e β são fórmulas e x é uma variável:

1. $\mathfrak{I}, \mu \models P(t_1, \dots, t_n)$ se e só se $\langle d_1, \dots, d_n \rangle \in \mathcal{P}$, onde $\mathcal{P} = \mathfrak{L}[P]$ e $d_i = \|t_i\|_{\mathfrak{I},\mu}$;
2. $\mathfrak{I}, \mu \models t_1 = t_2$ se e só se $\|t_1\|_{\mathfrak{I},\mu}$ e $\|t_2\|_{\mathfrak{I},\mu}$ são o mesmo elemento de \mathfrak{D} ;
3. $\mathfrak{I}, \mu \models \neg \alpha$ se e só se não se dá caso de $\mathfrak{I}, \mu \models \alpha$;
4. $\mathfrak{I}, \mu \models (\alpha \wedge \beta)$ se e só se $\mathfrak{I}, \mu \models \alpha$ e $\mathfrak{I}, \mu \models \beta$;
5. $\mathfrak{I}, \mu \models (\alpha \vee \beta)$ se e só se $\mathfrak{I}, \mu \models \alpha$ ou $\mathfrak{I}, \mu \models \beta$ ou ambos;
6. $\mathfrak{I}, \mu \models \exists x. \alpha$ se e só se $\mathfrak{I}, \mu' \models \alpha$ para uma varável de atribuição μ' que difere de μ no máximo até x ;
7. $\mathfrak{I}, \mu \models \forall x. \alpha$ se e só se $\mathfrak{I}, \mu' \models \alpha$ para toda varável de atribuição μ' que difere de μ no máximo até x .

Quando a fórmula α é uma frase a sua satisfação não depende da variável de atribuição, pois estas não contêm variáveis livres. Neste caso escrevemos $\mathfrak{I} \models \alpha$ o que significa que α é verdadeiro na interpretação \mathfrak{I} e falso caso contrário. No subconjunto proposicional da linguagem por vezes escrevemos $\mathfrak{L}[a] = 1$ ou $\mathfrak{L}[a] = 0$ de acordo com $\mathfrak{L} \models a$ seja verdadeiro ou não. A notação $\mathfrak{I} \models \mathcal{S}$ significa que as todas frases do conjunto \mathcal{S} são verdadeiras em \mathfrak{I} . Neste caso dizemos que \mathfrak{I} é o *modelo lógico* de \mathcal{S} .

7.2.1.3. PRAGMÁTICA

As regras de interpretação dizem-nos como entender precisamente o significado dos termos ou fórmulas da FOL em termos de domínio e interpretação para os símbolos não lógicos sobre o mesmo. O problema consiste em como esta linguagem representa o conhecimento. Pois é necessária uma interpretação inicial, ou seja, o sistema terá de ter uma interpretação que não parece simples.

7.2.1.3.1. CONSEQUÊNCIA LÓGICA

Esta noção permite relacionar as diversas ligações entre as frases da linguagem que não dependem do significado dos símbolos não lógicos. Embora as regras semânticas dependam da interpretação

destes símbolos. Sendo α e β duas frases da FOL e γ a frase $\neg(\beta \wedge \neg\alpha)$. Supondo que \mathfrak{I} é uma interpretação onde α é verdade. Utilizando as regras apresentadas facilmente observamos que γ deverá ser verdade, também, nessa interpretação. Desde que α seja verdade γ também será. Observamos que γ é implícito na verdade de α . Neste caso γ é a *consequência lógica* de α . Sendo mais preciso; seja S um conjunto de frases e α uma qualquer frase, afirmamos que α é uma *consequência lógica* de S , ou que S *implica logicamente* α . Escrevemos $S \models \alpha$ se e só se para *todas* as interpretações \mathfrak{I} , se $\mathfrak{I} \models S$ então $\mathfrak{I} \models \alpha$. Ou seja, cada modelo de S satisfaz α . Outra forma é que não existe uma interpretação \mathfrak{I} onde $\mathfrak{I} \models S \cup \{\neg\alpha\}$. Neste caso o conjunto $S \cup \{\neg\alpha\}$ diz-se que não é *satisfazível*.

Um caso especial desta definição é que uma frase α é *logicamente válida*, $\models \alpha$, quando é uma consequência lógica do conjunto vazio. Ou seja, α é válido se e só se para cada interpretação \mathfrak{I} se dá o caso $\mathfrak{I} \models \alpha$ ou, ainda, se e só se o conjunto $\{\neg\alpha\}$ não é satisfazível. Assim a validade é um caso especial de implicação, mas, também, de implicação quando o conjunto é finito: se $S \cup \{\alpha_1, \dots, \alpha_2\}$ então $\mathfrak{I} \models \alpha$ se e só se a frase $[(\alpha_1 \wedge \dots \wedge \alpha_2) \supset \alpha]$ é válida.

7.2.1.3.2. INFLUÊNCIA DA IMPLICAÇÃO LÓGICA

O fulcro da representação do conhecimento está na ligação entre o sistema baseado em conhecimento e a implicação lógica. O sistema que se pretende é um que consiga raciocinar. Ou seja, partir de uma determinada frase e concluir outras relações. Um sistema que acredita que uma frase é verdadeira numa interpretação e venha a crer noutras nessa mesma interpretação. Um sistema baseado no conhecimento não tem, nem pode ter, acesso à interpretação dos símbolos não lógicos. Pois poderia envolver um conjunto infinito de objetos fora do alcance de qualquer sistema computacional. Assim não poderá decidir com bases nas regras anteriores nem pode obter o conjunto de frases verdadeiras nessa interpretação, pois, haveria um conjunto infinito das mesmas. Contudo supondo um conjunto de frases S que implicam a frase α , sabemos que qualquer que seja a interpretação e se S for verdadeiro na interpretação então α também o é. Outras frases que não são implicadas também o podem ser, ou não, mas um sistema pode concluir quais as que são implicadas. Mas, aparentemente, o sistema não trás nada de novo pois as suas descobertas apenas apresentam as frases que já são implícitas no conhecimento. É necessário algo mais que a implicação através da consequência lógica. É necessário incluir no conjunto S um elemento que faça a ligação entre os símbolos não lógicos envolvidos. Sendo assim cada vez que introduzimos um elemento deste tipo, em S , retiramos interpretações possíveis do sistema. Estas interpretações podem ser indesejáveis e ao retirá-las tornamos o sistema mais compatível com a verdade na interpretação pretendida.

7.2.1.4. CRENÇA EXPLÍCITA E IMPLÍCITA

Á coleção de frases como premissas para utilização como base do cálculo da implicação é o que chamamos de *base de conhecimento*, como anteriormente já explorado. No caso presente o um conjunto

finito de frases da linguagem lógica de primeira ordem. O sistema de representação tem o papel de calcular as implicações da base de conhecimento. Podemos pensar como as crenças do sistema são explicitamente dadas e as implicações dessa base como as crenças que são implícitas. A este ponto falta, ainda, o raciocínio para que se consiga das crenças explícitas extrair aquelas implícitas.

7.2.1.4.1. EXEMPLO

Utilizando o exemplo do mundo de três blocos, ilustrado de seguida, vamos observar que a resposta carece de algum raciocínio. Supondo três blocos empilhados, Figura 7.1. O do topo é verde o do fundo não é verde e a cor do do meio não é conhecida. A pergunta centra-se em saber se existe um bloco verde diretamente por cima de um bloco não verde. Por observação é verdade. Mas chegar à resposta parece ser mais complexo do que esperado.

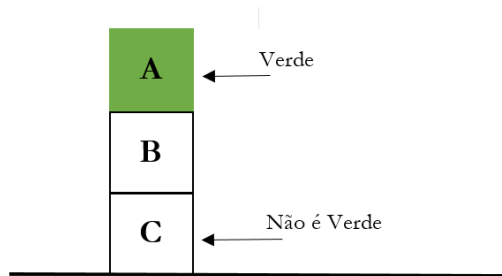


Figura 7.1 – Mundo de 3 blocos

Na FOL podemos formalizar o problema da seguinte forma: os factos de S são $[E(a, b), E(b, c), V(a), \neg V(c)]$. Sendo a, b e c os blocos e E e V significam “Em Cima” e “Verde” respetivamente. O que se afirma é que os factos implicam que existe um bloco verde em cima de um bloco não verde, ou seja, $S \models \alpha$ onde α é $\exists x \exists y, V(x) \wedge \neg V(y) \wedge E(x, y)$. Para isso é necessário que qualquer interpretação que satisfaça S também satisfaz α . Então, assumindo $\mathfrak{I} \models S$, existem dois casos a considerar:

1. Supondo $\mathfrak{I} \models V(b)$, então devido a $\neg V(c)$ e $E(b, c)$ estarem ambos em S temos $\mathfrak{I} \models V(b) \wedge \neg V(c) \wedge E(b, c)$ o que implica $\mathfrak{I} \models \exists x \exists y, V(x) \wedge \neg V(y) \wedge E(x, y)$;
2. Supondo que não é o caso, assim $\mathfrak{I} \models \neg V(b)$, então devido a $V(a)$ e $E(b, c)$ estarem ambos em S temos $\mathfrak{I} \models V(a) \wedge \neg V(b) \wedge E(b, c)$ o que implica $\mathfrak{I} \models \exists x \exists y, V(x) \wedge \neg V(y) \wedge E(x, y)$.

Em qualquer dos casos $\mathfrak{I} \models \alpha$ o que mostra que α é uma consequência lógica de S . Desta forma vemos que o cálculo dos elementos implícitos de um conjunto de dados pode envolver uma forma sutil de raciocínio. Geralmente, na FOL, a determinação de se uma frase é consequência lógica de outras é um problema *insolúvel*. Nenhum processo automático pode decidir da validade e nenhum processo automático poderá dizer-nos em todos os casos se uma frase é ou não implicada.

7.2.1.4.2. SISTEMAS BASEADOS EM CONHECIMENTO

Uma KB contém uma quantidade de informação explícita. O sistema deverá ter um comportamento o mais próximo possível do que é implícito pela sua base. Este processo irá requer raciocínio. A

inferência dedutiva é o processo de calcular as implicações da base de conhecimento, isto é, dada a KB e qualquer frase α determinar $KB \models \alpha$.

Um processo de raciocínio é considerado *logicamente correto* se ao produzir α este é garantidamente uma consequência lógica. Previne a produção de assunções plausíveis que podem ser verdade na interpretação, mas não são estritamente implicadas. Um processo de raciocínio é considerado *logicamente completo* ao produzir α sempre que este é implicado. Previne a perda de implicações quando o seu estado é complicado de determinar.

7.3. EXPRESSÃO DO CONHECIMENTO

A base de conhecimento compreende um conjunto frases e as proposições expressas por essas são as crenças do possível agente. Essas frases pretendem satisfazer alguma propriedade o mundo apresentado. Da mesma forma que um programador pensa na arquitetura do programa que suporta o seu projeto é, da mesma forma, necessário compreender a arquitetura do sistema. É necessário acomodar a estrutura do sistema e o que servirá como o objetivo final. Sendo assim define-se as ocasiões que a inferência será necessária bem como o que o sistema irá computar. Depois dessas decisões criamos algo, por vezes chamado de *ontologia*, que é a definição dos tipos de objetos importantes, quais as suas propriedades e as relações entre eles. Tudo isto antes de popular a base de conhecimento. Este processo, que trabalha na base de conhecimento ao nível deste, é chamado, por vezes, de *engenharia do conhecimento*. O objetivo é criar uma base de conhecimento que contém as implicações apropriadas e dos primeiros pontos a trabalhar é o vocabulário para utilizar e que factos representar. Na criação de uma KB devemos começar com o conjunto de predicados dependentes do domínio e funções que providenciam a base para afirmação dos factos sobre o domínio. Determinar todos os elementos centrais do mundo e depois circunscrever os tipos básicos de objetos desses elementos. Seguidamente definir os atributos cruciais desses objetos. As relações entre esses objetos são as primeiras as serem consideradas e depois algumas relações mais complexas que possam ser importantes. Por fim definimos as funções importantes do domínio. Estas podem tomar mais do que um argumento bem como as relações poderão ter várias aridades.

Assim sendo podemos começar a referir os factos existentes. Aplicando as frases atómicas e negação das mesmas para aplicar aos tipos de predicados para representar verdades básicas. Estas verdades poderiam definir uma ontologia básica para o mundo. Após a definição dos objetos trataríamos da definição das suas propriedades. Estas serão o centro do conhecimento a representar pois, a partir das implicações do conjunto de factos ou conjeturas, representarão o conhecimento que queremos extrair da KB . Estes factos básicos podem ser comparados a uma pequena base de dados, sendo cada predicado comparado uma tabela e os registos aos identificadores de todos os elementos que satisfazem o predicado. A estratégia de armazenamentos seria a um nível simbólico. Um outro conjunto de factos importantes na representação do domínio seria a igualdade/atribuição. Estes

factos seriam para definir relação entre elementos que poderiam não ter propriedades comuns, mas, mesmo assim, possuir um determinado tipo de relação.

Todos estes factos apenas cobrem os aspetos mais básicos. Muitos dos factos a expressar são mais complexos do que aqueles que é possível representar através de simples frases atómicas. Assim surge a necessidade da utilização de fórmulas mais complexas, com quantificadores e outros conectores, para expressar as várias crenças sobre o domínio. Os quantificadores universais são úteis pois permitem uma espécie de abreviatura na expressão dos factos. Poderíamos enumerar todos os determinados elementos relativos a uma propriedade, mas, se estes fossem muitos, seria demasiado complexo. Além de que a introdução de novos elementos seria já suportada pelos quantificadores universais. Em contraste existem fórmulas para expressar o conhecimento incompleto sobre o mundo. Neste caso os quantificadores universais não são uma abreviação, mas sim simplesmente a expressão de algo que não se conhece. Não é possível escrever uma expressão mais completa.

Um outro facto complexo a ter em conta é a chamada *frase de encerramento* ou *frase de fecho* utilizada como limitação do domínio do discurso. Este facto permite que não seja postulado um novo elemento durante o processo de raciocínio. Por fim é necessário distinguir formalmente todos os elementos do domínio para que não se dê o caso de postular dois elementos distintos como sendo o mesmo. Todos estes factos servem para representar as circunstâncias básicas do domínio e um arranque para o processo de raciocínio. Para relacionar os predicados e símbolos de funções utilizados é necessário ter atenção a elementos que nos permitam inferir algumas relações óbvias dentro de um domínio. Por exemplo definimos uma pessoa como homem é lógico que não será uma mulher. Mas é necessário fornecer ao sistema suporte para tal. Estes destinam-se a clarificar a terminologia utilizada.

Como exemplos:

- **Disjunção:** muitas vezes dois predicados são disjuntos e a asserção de um implica a negação de um outro: $\forall x [Homem(x) \supset \neg Mulher(x)]$;
- **Subtipos:** existem predicados que podem implicar um tipo de especialização, por exemplo um determinado tipo pode conter outro, como a relação entre médico e cirurgião: $\forall x [Cirurgião(x) \supset Médico(x)]$. Assim é razoavelmente possível inferir que tudo o que é verdade para um médico é verdade para o cirurgião, mas não contrário;
- **Exaustão:** é como o inverso do interior onde dois, ou mais, subtipos formam o supertipo: $\forall x [Adulto(x) \supset (Homem(x) \cup Mulher(x))]$;
- **Simetria:** relações simétricas: $\forall x, y [CasadoCom(x, y) \supset CasadoCom(y, x)]$;
- **Inverso:** outras são opostas: $\forall x, y [FilhoDe(x, y) \supset PaidDe(y, x)]$;
- **Restrições de Tipo:** os argumentos devem ser de um determinado tipo: $\forall x, y [CasadoCom(x, y) \supset Pessoa(x) \wedge Pessoa(y)]$;
- **Definição Completa:** criar um predicado composto definido pela combinação lógica de outros predicados: $\forall x [Homem(x, y) \equiv Rico(x) \wedge Homem(x)]$.

Como estes exemplos mostram que os factos terminológicos são, tipicamente, capturados através de uma linguagem lógica, como condicionais quantificados ou bicondicionais.

Nesta altura a estrutura básica está criada faltando, agora, o que é o objetivo final do sistema: derivar conclusões implícitas a partir da representação explícita na base de conhecimento. Não apenas questões simples, mas, também as mais complexas. É necessário determinar se a verdade implícita por uma frase é implicada pela base KB . Para isso é necessário determinar se cada interpretação lógica que satisfaz a base de conhecimento também satisfaz a frase. Por vezes esta determinação trás à superfície informação que não era procurada. O que evidencia a possibilidade de responder não só a informação verdadeira-falsa, mas também para extrair outro tipo de informação não binária. Assim o sistema permite procurar também outra informação. Um ponto importante é quando pretendemos satisfazer a validade de uma frase quando a resposta é não. Para o demonstrar é necessária uma interpretação que satisfaz a KB , mas falsifica a frase. Isto é dada uma interpretação $\mathfrak{I} = \langle \mathcal{D}, \mathcal{Q} \rangle$ argumentar que satisfaz cada frase na KB bem com a negação da frase. Isto implica um grande processo pois cada frase da KB tem de ser verificada, mas, basicamente, significa que não pode contradizer o que já está presente.

A FOL fornece ferramentas para representar factos num domínio. Mas em muitos casos existe uma grande flexibilidade que pode ser exercida no mapeamento objetos no domínio em predicados e funções. Também existe considerável flexibilidade no que consideramos os elementos do domínio. Por vezes é útil introduzir *elementos abstratos* que podem não ter sido considerados na primeira análise. Esta ideia é chamada de *reificação* e é típica em sistema de lógica e plataformas de linguagens. Esta observação torna-se mais pertinente quando se tenta expressar mais detalhes do elemento. Estes detalhes aumentam a aridade do predicado. Pelo que a criação de elementos abstratos para tratar destes casos torna-se evidente. Assim a conjunção destes predicados dará o nível de detalhe desejado e, estes, podem ser definidos inicialmente. Vários exemplos destes elementos abstratos apresentam-se-nos facilmente: números, datas, tempos, endereços entre outros. Depois destas observações teremos um sistema que representa os factos básicos e elementos de um domínio com *sensu commum* do mundo a representar. Mas existem outros tipos de factos que podemos querer capturar e que podem ser problemáticos através da FOL. Embora possam ser capturados através de extensões da linguagem. Alguns exemplos destes tipos de factos:

- Factos estatísticos e probabilísticos;
- Factos por definição e protótipos;
- Factos intencionais.

Todos estes elementos permitem-nos criar uma KB para representar o domínio de um mundo.

7.4. REPRESENTAÇÃO ORIENTADA A OBJETOS

A representação descrita até agora tem uma propriedade comum: é plana. Ou seja, cada elemento de informação é fechado e pode ser compreendido independentemente dos outros. Como vimos a informação sobre um determinado elemento pode estar dispersa na base de conhecimento. Com o aumento dos elementos é crítico que se possa organizar estes elementos da KB . Uma aproximação

que respeite mais o nosso objetivo de representação do conhecimento será agrupar os factos e regras em termos dos elementos a que se referem. Os objetos que nos podem interessar são muitos e variados. Desde objetos físico até objetos abstratos e ainda abstrações reificadas como eventos e relações. Todos estes objetos apresentam algumas partes, atributos, que por sua vez também ser abstratos. Estas partes, por sua vez podem apresentar várias restrições. Podemos pensar que um telhado tem de estar ligado de uma forma às paredes ou mesmo relações abstratas como a primeira parte de uma viagem tem de ser realizada em primeiro lugar que a segunda. Estas restrições podem ser expressas de forma procedimental. Em geral consideramos, nos sistemas de conhecimento com objetos, as operações de raciocínio que são relevantes para os tipos de objetos que interessam. Marvin Minsky, em 1975, sugeriu a ideia de utilizar grupos orientados a objetos de procedimentos para reconhecer e lidar com novas situações. Apesar de ter um determinado objetivo a sua ideia obteve uma grande aplicação na representação do conhecimento. A *frame*, Figura 7.2, foi o elemento utilizado por Minsky para a estrutura de dados para representar estas situações. Genericamente existem dois tipos de *frames*. As individuais que representam simples objetos e genéricas para representar categorias ou classes de objetos. Uma *frame* individual é uma lista de elementos onde colocamos os valores, estes elementos são chamados de *slots* e os itens lá colocados designam-se por *fillers*.

```
(NomeDaFrame
  <NomeDaSlot1 filler1>
  <NomeDaSlot2 filler2>
  ...)
```

Figura 7.2 – *Frame* individual

A *frame* e os *slots* são símbolos atômicos; os *fillers* ou são valores individuais ou nomes de outras *frames* individuais. Estas por sua vez contêm um *slot* especial, **:INSTANCE-OF**, cujo *filler* é a identificação de uma *frame* genérica, Figura 7.3, que indica a categoria do objeto. Estas *frames*, na sua constituição mais básica, têm uma sintaxe semelhante a *frames* individuais.

```
(CidadePortuguesa
  <:IS-A Cidade>
  <:Região RegiãoPortuguesa>
  <:País Portugal>)
```

Figura 7.3 – *Frame* genérica

Os *slot fillers* são a identificação de *frames* genéricas ou individuais. E contêm um *slot* **:IS-A** cujo *filler* é o nome de uma *frame* ainda mais genérica. Assim as *frames* genéricas podem ser especializações de *frames* ainda mais genéricas. Os *slots* podem ter *attached procedures*, por exemplo **IF-ADDED** e **IF-NEEDED**, e *fillers* conjuntamente na mesma *frame*.

Muito do raciocínio neste sistema envolve a criação de instâncias individuais de *frames* genéricas, preenchimentos de *slots* com valores e inferência de outros. Os *slots* **:INSTANCE-OF** e **:IS-A** têm um papel especial no processo pois, em particular, as *frames* genéricos podem ser utilizados para encontrar valores que não foram explicitamente mencionados na criação da instância e, também,

disparar ações adicionais quando os *fillers* são providenciados. O processo de passar informação da *frame* genérica para as suas especializações e, eventualmente, para as suas instâncias é chamado de *herança*. Mesmo que não se indique *filler* para os *slots* através da herança saberemos informação genérica sobre a instância. Os *attached procedures* funcionam da mesma forma com o processo de herança. Convém comentar que a utilização dos elementos genéricos, da herança, entram em jogo quando não fornecemos os *fillers* para os *slots*. Isto provê a capacidade de, nas suas instâncias e especializações, os valores serem substituídos pelos mais indicados. Também é possível as *frames* individuais serem instâncias de mais de que uma *frame* genérica e estas serem especializações de outras ainda mais genéricas.

O raciocínio através de *frames* geralmente inicia-se com o sistema a reconhecer um objeto como instância de uma *frame* genérica e então aplica os procedimentos ativados por aquele reconhecimento. Isto é possível porque os procedimentos suportam uma plataforma flexível para a computação. O processo de raciocínio básico num sistema de *frames* contém os três passos seguintes:

1. O sistema, utilizando as *frames*, declara que existe um objeto ou situação despoletando a instanciação de uma *frame* genérica;
2. Cada *filler* não explicitamente providenciado, mas capaz de ser herdado da *frame* genérica é herdado pela nova instância;
3. Para cada *slot* com *filler* o procedimento **IF-ADDED**, que pode ser herdado, é despoletado, possivelmente criando novos *slots* a serem preenchidos ou novas *frames* a instanciar e o ciclo repete-se.

Se o sistema, utilizador ou procedimento requiere o *filler* de um *slot* o seguinte comportamento ocorre:

1. Se existe um *filler* no *slot* o valor é retornado;
2. Caso contrário o procedimento **IF-NEEDED**, que pode ser herdado, é despoletado calculando o *filler* para o *slot* mas, potencialmente, provocando que outros *slots* sejam preenchidos ou novas *frames* instanciadas.

Se nenhum resultado é produzido o valor é considerado desconhecido. De notar que a herança de valores é feita na altura que a *frame* é instanciada, mas os procedimentos são invocados apenas quando necessários. Este processo é o processo de raciocínio envolvendo uma *frame* localmente. Numa plataforma que sustenta uma base de conhecimento baseada em *frames* é necessário pensar na estrutura global do sistema e como a computação deverá produzir os resultados esperados do raciocínio. Geralmente as *frames* genéricas são criadas para cada tipo de objeto ou de situação para a resolução do problema. As restrições entre os *slots* são expressas e executadas pelos procedimentos. Isto permite que os procedimentos na *frame* da base de conhecimento suportem ações mais robustas, como invocar ações complexas pelo sistema.

Existe uma troca de valores entre os tipos de procedimentos utilizados, irá depender do peso dado aos valores computados de início contra o valor de esperar pelos resultados computados apenas quando necessário. Como já feito para a linguagem lógica, também, foram feitas extensões para suportar aplicações mais complexas. Como outros tipos de procedimentos, múltiplos *fillers* para os *slots*, outros usos dos *slots*, *metaframes*.

A utilização do sistema com *frames* pode aproximar-se da programação orientado a objetos dependendo do uso dado ao sistema. Se nos deslocarmos para a visão mais procedimental das *frames* teremos uma base de conhecimento mais procedimental. Como todas as vantagens e desvantagens inerentes. Um sistema em *frames* tem um fulcro mais centralizado, trabalhando num ciclo: instanciação de uma *frame* e declaração dos *fillers*, herança de valores, disparo dos procedimentos e consequentes ações e depois de estabilizar a espera por nova entrada. Os sistemas por objetos tendem a ser mais descentralizados com cada um agindo como agente independente enviando mensagens entre si. [45]

7.5. DESCRIÇÕES ESTRUTURADAS

No ponto anterior discutimos como organizar a informa em termos de objetos. Uma tendência naturalmente humana. Mas, como vimos, a representação por *frames* parece focar-se na organização e invocação de procedimentos em vez de inferências sobre os objetos e categorias [45]. O raciocínio sobre objetos diários vai mais além do que ilustrado, como por exemplo:

- Os objetos, geralmente, pertencem a uma categoria, mas por vezes a várias;
- As categorias podem ser mais gerais ou particulares;
- A generalização é comum para categorias com nomes simples, mas também para as com mais complexas descrições;
- Os objetos contêm partes, e por vezes múltiplas;
- As relações entre as partes de um objeto são essenciais para considerar o objeto como membro da categoria.

Anteriormente trabalhamos com frases na linguagem lógica pois estas representam o que se conhece. Vamos tratar, agora, de *frases substantivas* que podem ser simples ou complexas. Num capítulo anterior discutimos os predicados. Consideramos que os mesmos representavam as categorias através de uma palavra. Podemos verificar que para capturar tipos mais interessantes necessitamos de predicados com uma estrutura interna. A um predicado verdadeiramente composto, por exemplo com dois nomes, é expectável que para cada elemento individual do predicado seja verdadeiro se o composto também o é. Esta solução não é verdade para factos do mundo, mas por definição. Este processo ocorre com naturalidade na linguagem natural. Na linguagem lógica de primeira ordem natural não existem ferramentas que tratem deste aspeto. Mas a prevalência desta construção na linguagem torna interessante o trabalho de criar ferramentas que suportem esta representação. A lógica desta representação trata de manipular os predicados complexos através de *descrições*, este sistema lógico designa-se por *lógica de descrição – Description Logic (DL)*.

Podemos imediatamente observar que existem dois tipos de elementos: os elementos que representam as classes básicas de objetos e elementos que são parte, atributo ou propriedade de outros objetos. O primeiro tipo é denominado como *conceito* e o segundo como *papel*. Será semelhante à distinção entre *frame* e *slot* apresentados anteriormente. Como acontecia com as *frames* vamos considerar os conceitos organizados hierarquicamente por generalização. Mas nesta linguagem muito do processo de generalização segue logicamente do significado dos elementos envolvido no

predicado. Não hierarquizado por intervenção do utilizador. Assim o processo de raciocínio irá centrar-se muito na computação automática desta relação de generalização. Por simplificação os papéis não são considerados como tendo esta organização hierárquica, mas poderão ter múltiplos *fillers*. Também consideramos indivíduos, para aplicar estas descrições, como tal a linguagem incluirá *constantes*.

7.5.1. LINGUAGEM DE DESCRIÇÃO

De [45] obtemos uma sintaxe para uma linguagem de descrição simples chamada \mathcal{DL} – *Description Language*. Contém dois tipos de símbolos: lógicos com um significado e/ou uso fixo e não lógicos que são dependentes da aplicação. São quatro, os conjuntos, de símbolos lógicos:

- *Pontuação*: [,] , (e);
- *Inteiros positivos*: 1, 2, 3, ...;
- *Operadores de formação de conceitos*: **ALL**, **EXISTS**, **FILLS** e **AND**;
- *Conectores*: \sqsubseteq , \doteq e \rightarrow .

Na linguagem existem três tipos de símbolos não lógicos:

- *Conceitos atômicos*: capitulação mista; um conceito atômico especial *Thing*;
- *Papel*: como os conceitos atômicos, mas antecidos por “:”;
- *Constantes*: sem capitulação.

Existem quatro tipos legais de expressões na linguagem: constantes, papéis, conceitos e frases. As letras c e r representam constantes e papéis respetivamente. Os conceitos são representados por d e e ficando os conceitos atômicos representados por a . O conjunto de conceitos da \mathcal{DL} é o conjunto mínimo que satisfaz o seguinte:

- Todo o conceito atômico é um conceito;
- Se r é um papel e d é um conceito então $[\mathbf{ALL} \ r \ d]$ é um conceito;
- Se r é um papel e n é um inteiro positivo $[\mathbf{EXISTS} \ n \ r]$ é um conceito;
- Se r é um papel e c é uma constante então $[\mathbf{FILLS} \ r \ c]$ é um conceito;
- Se $d_1 \dots d_n$ são conceitos então $[\mathbf{AND} \ d_1 \dots d_n]$ é um conceito.

Existem três tipos de frases:

- Se $d_1 \dots d_n$ são conceitos então $(d_1 \sqsubseteq d_n)$ é uma frase;
- Se $d_1 \dots d_n$ são conceitos então $(d_1 \doteq d_n)$ é uma frase;
- Se c é uma constante e d um conceito então $(c \rightarrow d)$ é uma frase.

Uma KB numa \mathcal{DL} é considerada como uma qualquer coleção de frases nesta forma. As constantes representam os indivíduos no domínio da aplicação, como na FOL, os conceitos categorias e classes e os papéis as relações binárias entre esses indivíduos. Os conceitos complexos são derivados do significado das suas partes. O conceito $[\mathbf{EXISTS} \ n \ r]$ representa a classe de indivíduos no domínio que estão relacionados, através da relação r , com, pelos menos, n outros indivíduos. Já o conceito $[\mathbf{FILLS} \ r \ c]$ representa os indivíduos que estão relacionados, através de r , com o indivíduo c . Considerando d como uma determinada classe $[\mathbf{ALL} \ r \ d]$ representa os elementos que estão

relacionados através de r apenas com os elementos classe d . Por fim o conceito $[\text{AND } d_1 \dots d_n]$ representa os elementos que se são descritos por $d_1 \dots d_n$.

As frases são esperadas que sejam verdadeiras ou falsas no domínio. Considerando os conceitos d_1 e d_2 , representando duas classes de indivíduos, e a constante c que representa um indivíduo. Então $(d_1 \sqsubseteq d_2)$ indica que o conceito d_1 é subsumido pelo conceito d_2 , isto é, todos os indivíduos que satisfazem d_1 também o fazem relativamente a d_2 , por exemplo (*Cirurgião* \sqsubseteq *Médico*). Significa que um cirurgião é também um médico, entre outras coisas. De forma similar $(d_1 \doteq d_n)$ significa que ambos os conceitos são equivalentes, ou seja, os indivíduos que satisfazem d_1 são precisamente aqueles que satisfazem d_2 . É uma forma reduzida de afirmar $(d_1 \sqsubseteq d_2)$ e $(d_2 \sqsubseteq d_1)$ são ambas verdadeiras. A frase $(c \rightarrow d)$ representa o indivíduo denotado por c que satisfaz a descrição expressa pelo conceito d . Todas as frases em \mathcal{DL} são atômicas pelo que é simples criar conceitos mais complexos. Uma frase típica na KB na \mathcal{DL} é a atribuição de um nome a um conceito complexo. A utilização do conector \doteq indica um conceito equivalente à descrição apresentado à direita, ou seja, é *definido* totalmente na KB . A não utilização de \doteq na KB apresenta o conceito como subsumido, *conceito primitivo*, porque temos as condições necessárias a satisfazer. Ao contrário do conector \doteq que apresenta as condições suficientes e necessárias.

7.5.2. SIGNIFICADO E IMPLICAÇÃO

O ponto de início da semântica da descrição lógica é a interpretação, como já visto na FOL. A interpretação \mathfrak{I} para \mathcal{DL} é um par $\langle \mathfrak{D}, \mathfrak{L} \rangle$, como já descrito, onde \mathfrak{D} é um conjunto de objetos, domínio da interpretação, e \mathfrak{L} é o mapeamento da interpretação dos símbolos não lógicos de \mathcal{DL} para os elementos e relações sobre \mathfrak{D} , onde:

- Para cada contante c , $\mathfrak{L}[c] \in \mathfrak{D}$;
- Para cada conceito atômico a , $\mathfrak{L}[a] \subseteq \mathfrak{D}$;
- Para cada papel r , $\mathfrak{L}[r] \subseteq \mathfrak{D} \times \mathfrak{D}$.

Comparando com a FOL vemos que as contantes apresentam o mesmo significado dos termos, os conceitos atômicos são compreendidos como predicados unários e os papéis como predicados binários. O conjunto $\mathfrak{L}[d]$ associado com o conceito d numa interpretação é designado por *extensão* e \mathfrak{L} é, por vezes, chamado função de extensão. Uma característica diferenciadora da é a presença de conceitos não atômicos em que o seu significado é completamente determinado pelo significado das partes constituintes. Mais genericamente é possível estender a definição de \mathfrak{L} a todos conceitos como mostrado de seguida:

- Para o conceito especial *Thing*, $\mathfrak{L}[\textit{Thing}] = \mathfrak{D}$;
- $\mathfrak{L}[[\text{ALL } r \ d]] = \{x \in \mathfrak{D} \mid \text{para qualquer } y, \text{ se } (x, y) \in \mathfrak{L}[r], \text{ então } y \in \mathfrak{L}[d]\}$;
- $\mathfrak{L}[[\text{EXISTS } n \ r]] = \{x \in \mathfrak{D} \mid \text{existe pelo menos } n \text{ distintos } y \text{ tal que } (x, y) \in \mathfrak{L}[r]\}$;
- $\mathfrak{L}[[\text{FILLS } r \ c]] = \{x \in \mathfrak{D} \mid (x, \mathfrak{L}[d]) \in \mathfrak{L}[r]\}$;

- $\mathfrak{L}[\text{[AND } d_1 \dots d_n]] = \mathfrak{L}[d_1] \cap \dots \cap \mathfrak{L}[d_n]$.

Dadas as corretas condições estas regras permitem-nos encontrar a extensão de qualquer conceito.

Dada uma interpretação \mathfrak{I} é possível especificar as frases de \mathcal{DL} que são falsas, ou verdadeiras, conforme a interpretação. A frase $(c \rightarrow d)$ será verdadeira quando o objeto denotado por c está na extensão de d ; a frase $(d \sqsubseteq d')$ será verdade quando a extensão de d é um subconjunto da extensão de d' ; a frase $(d \doteq d')$ será verdade quando a extensão de d é a mesma que a extensão de d' .

Formalmente dada uma interpretação $\mathfrak{I} = \langle \mathcal{D}, \mathfrak{L} \rangle$ dizemos que α é verdade em \mathfrak{I} , $\mathfrak{I} \models \alpha$, de acordo com as seguintes regras, assumindo que d e d' são conceitos e c é uma constante:

- $\mathfrak{I} \models (c \rightarrow d)$ se e só se $\mathfrak{L}[c] \in \mathfrak{L}[d]$;
- $\mathfrak{I} \models (d \sqsubseteq d')$ se e só se $\mathfrak{L}[d] \subseteq \mathfrak{L}[d']$;
- $\mathfrak{I} \models (d \doteq d')$ se e só se $\mathfrak{L}[d] = \mathfrak{L}[d']$.

Como anteriormente utilizamos a notação $\mathfrak{I} \models \mathcal{S}$, onde \mathcal{S} é um conjunto de frases, para significar que todas as frases em \mathcal{S} são verdade em \mathfrak{I} . A definição de implicação lógica é exatamente como definido para a FOL. Sendo \mathcal{S} um conjunto de frases e α uma frase dizemos que \mathcal{S} implica logicamente α $\mathcal{S} \models \alpha$, se e só se para cada interpretação \mathfrak{I} se $\mathfrak{I} \models \mathcal{S}$ então $\mathfrak{I} \models \alpha$. Como caso especial da definição temos que a frase α é logicamente válida, denotada por $\models \alpha$, quando é logicamente implicada pelo conjunto vazio. Existem dois tipos de raciocínio importantes: determinar se uma constante c satisfaz um certo conceito d e determinar se um conceito d é ou não subsumido pelo conceito d' . Estes processos envolvem o cálculo das implicações da base de conhecimento.

Dada a base de conhecimento KB é necessário determinar se $KB \models \alpha$ para frases α na forma:

- $(c \rightarrow d)$ onde c é uma constante e d um conceito; e
- $(d \sqsubseteq e)$ onde ambos d e e são conceitos – $KB \models (d \doteq e)$ se e só se $KB \models (d \sqsubseteq e)$ e $KB \models (e \sqsubseteq d)$.

Repara-se que o primeiro ponto depende de se ser capaz de suportar o segundo. Dessa forma é necessário considerar a computação da subsunção. Este processo pode ser resumido brevemente:

- **Simplificação da base de conhecimento:** As implicações da subsunção não são afetadas pela presença de frases na forma $(c \rightarrow d)$ na KB. Ou seja, KB' é como KB excetuando que todas as frases $(c \rightarrow d)$ foram removidas. Podemos remover as frases na forma $(d \sqsubseteq e)$ substituindo-as por frases na forma $(d \doteq [\text{AND } e \ a])$ onde a é um novo conceito atômico, não utilizado em outro lado.
- **Normalização:** São realizadas inferências em pequena escala, um conceito de cada vez e em pequenos passos:
 - **Expansão das definições:** cada conceito atômico é substituído pela sua definição (conceito atômico na posição direita de uma frase \doteq);
 - **Nivelação dos operadores AND** cada sub conceito na forma $[\text{AND } \dots [\text{AND } d_1 \dots d_n] \dots]$ simplifica para $[\text{AND } \dots d_1 \dots d_n \dots]$;
 - **Combinação dos operadores ALL:** cada sub conceito na forma $[\text{AND } \dots [\text{ALL } r \ d_1] \dots [\text{ALL } r \ d_1] \dots]$ simplifica para $[\text{AND } \dots [\text{ALL } r [\text{AND } d_1 \ d_2]] \dots]$;

- **Combinação dos operadores EXISTS:** cada sub conceito na forma $[\mathbf{AND} \dots [\mathbf{EXISTS} \ n_1 \ r] \dots [\mathbf{EXISTS} \ n_2 \ r] \dots]$ simplifica para $[\mathbf{AND} \dots [\mathbf{EXISTS} \ n \ r] \dots]$, onde n é o máximo entre n_1 e n_2 ;
 - **Considerar [Thing]:** só deverá aparecer quando toda a expressão se reduz a si;
 - **Remover expressões redundantes:** eliminar todas a expressão que são exatamente o duplicado de outra dentro da mesma expressão **AND**.
- A normalização percorre estas operações repetidamente até terminar quando mais nenhum passo pode ser efetuado. No final o resultado é $[\mathbf{Thing}]$, um conceito atômico ou um conceito na seguinte forma:

$$\begin{aligned}
 & [a_1 \dots a_m \\
 & [\mathbf{FILLS} \ r_1 \ c_1] \dots [\mathbf{FILLS} \ r_{m'} \ c_{m'}] \\
 & [\mathbf{EXISTS} \ n_1 \ s_1] \dots [\mathbf{EXISTS} \ n_{m''} \ s_{m''}] \\
 & [\mathbf{ALL} \ t_1 \ e_1] \dots [\mathbf{ALL} \ t_{m'''} \ e_{m'''}]
 \end{aligned}$$

onde a_i são conceitos primitivos atômicos além de $[\mathbf{Thing}]$, e r_i , s_i , e t_i são papéis, c_i são constantes, n_i inteiros positivos e e_i são conceitos normalizados.

Para computar se $KB \models (d \sqsubseteq e)$ é necessário comparar as versões normalizadas de d e e . Esta ideia é que para que d seja subsumido por e , a normalização de d deve ter em atenção todos os elementos de e normalizado. A correção do processo de subsunção é mostrada por $KB \models (d \sqsubseteq e)$ (de acordo com a definição em termos da interpretação) se e só d normaliza para um d' , e e normaliza para um e' e para cada componente de e' existe um componente correspondente de d' . A computação se um indivíduo denotado por uma constante satisfaz um conceito é similar ao processo da subsunção entre dois conceitos. A maior distinção é que as frases \rightarrow na KB têm de ser tidas em contas. Ou seja, precisamos determinar se $KB \models (b \rightarrow e)$ ou não, onde b é uma constante e e é um conceito. Geralmente para determinar se um indivíduo satisfaz uma descrição é necessário *propagar* a informação implicada pelo que sabemos sobre outros indivíduos antes de verificar a subsunção. O procedimento, considerando que não existem termos **EXISTS** nos conceitos será o seguinte:

- Construir uma lista S de pares (b, d) onde b é uma constante da KB e d é versão normalizada do **AND** de todos os conceitos d' tais que $(b \rightarrow d')$ está na KB ;
- Procurar duas constantes, b_1 e b_2 , tais que (b_1, d_1) e (b_2, d_2) estão em S e para um papel r , $[\mathbf{FILLS} \ r \ b_2]$ e $[\mathbf{ALL} \ r \ e]$ são componentes de d_1 , mas não é o caso que $KB \models (d_2 \sqsubseteq e)$;
- Se nenhum par é encontrado então sair. Caso contrário substitui o par (b_2, d_2) em S por (b_2, d'_2) onde d'_2 é a versão normalizada de $[\mathbf{AND} \ d'_2 \ e]$ e voltar ao passo anterior.

O procedimento produz o efeito de computar para cada constante b o conceito mais específico b tal que $KB \models (b \rightarrow d)$. Após o processo ter corrido para testar se $KB \models (b \rightarrow d)$ ou não simplesmente testamos se $KB \models (d \sqsubseteq e)$. Para considerar os termos na forma $[\mathbf{EXISTS} \ 1 \ r]$ uma ideia semelhante pode ser utilizada. Temos agora como pares (b, σ, d) onde σ é uma cadeia de papéis e, intuitivamente, $b.r_1.r_2$ pode ser entendido como um indivíduo que é um r_2 de um r_1 de b . Quando σ é vazio isto corresponde a b . A extensão do *encadeamento* do procedimento a estes novos termos é feita com novos passo adicionais como se mostra:

- Procurar uma constante b , uma cadeia de papéis σ (talvez vazia) e um papel r tal que (b, σ, d_1) está em S e um qualquer (b, σ, r, d_2) existente em S (se não existir considerar d_2 como *Thing*) e onde $[EXISTS\ 1\ r]$ e $[ALL\ r\ e]$ são ambos componentes de d_1 , mas não é o caso que $KB \models (d_2 \sqsubseteq e)$;
- Se estes são encontrados remover o par (b, σ, r, d_2) de S (se exequível) e adicionar o par (b, σ, r, d'_2) onde d'_2 é a versão normalizada de $[ALL\ d_2\ e]$;
- Repetir.

Este processo estende a propagação a indivíduos anónimos, podemos começar com uma propriedade do indivíduo b, σ e concluir algo novo sobre o indivíduo b, σ, r . Eventualmente poderemos descobrir algo de novo sobre um indivíduo devidamente identificado. No caso dos termos $[EXISTS\ n\ r]$ observamos que não é necessário criar n diferentes indivíduos anónimos, porque iriam ter as mesmas propriedades na propagação. O caso mais geral considera $n = 1$. [45]

7.5.3. TAXONOMIA E CLASSIFICAÇÃO

Existe um número pequeno de questões que tipicamente seriam utilizadas neste sistema. Estas KB lembram bases de dados onde os conceitos correspondem aos elementos do esquema e as constantes a registos. Dessa forma é comum perguntar pelas instâncias de um conceito: seja um conceito q , encontrar todos os c em KB tal que $KB \models (c \rightarrow q)$. Por outro lado, lembram os sistemas em *frames*, é comum perguntar por todas as categorias que um indivíduo satisfaz para, por exemplo, acionar procedimentos associados às classes: dada a constante c , encontrar todos os conceitos atómicos a tais que $KB \models (c \rightarrow a)$. Desta forma poder-se-á pensar na realização de uma pesquisa sobre toda a KB . Esta irá crescer linearmente no tempo conforme o aumento de frases na KB . Não podemos esquecer que utilização da descrição lógica permite pensar nos conceitos como organizados hierarquicamente com os conceitos mais gerais no topo e os mais específicos no fundo. Considerando uma estrutura de dados em árvore o tempo de pesquisa cresce linearmente com a *profundidade* da taxonomia em vez do seu tamanho, o que permitirá considerar KB de maiores dimensões. A subsunção é uma ordem parcial e uma taxonomia naturalmente segue um conjunto de conceitos. Assumindo que o conjunto (a_1, \dots, a_n) de conceitos atómicos estão todos à esquerda das frases, \doteq e \sqsubseteq , na KB . A taxonomia resultante terá nós para cada a_i e arestas de a_i para a_j sempre que $a_i \sqsubseteq a_j$ e não existe nenhum a_k distinto tal que $a_i \sqsubseteq a_k \sqsubseteq a_j$. Isto produz um grafo direto acíclico e não terá ligações redundantes e a transitividade das ligações irão capturar todas as relações de subsunção implicadas pelas declarações definindo a_i . Adicionando o requisito que a constante c na KB seja ligada apenas ao mais específico conceito atómico a_i tal que $KB \models (c \rightarrow a_i)$ teremos uma representação hierárquica da KB .

Após a taxonomia definida podemos acrescentar frases para um novo conceito atómico ou constante. Irá envolver a criação de novas ligações para os novos elementos e, talvez, ao redireccionamento de ligações existentes. Este processo é chamado de *classificação*, este explora a estrutura de taxonomia

permitindo um funcionamento mais eficiente. A construção da taxonomia poderá ser feita através de classificação; adicionando ao conceito *Thing*, na taxonomia, novos conceitos e constantes incrementalmente.

O processo de classificação começa por adicionar a frase ($a_{\text{nov}} \doteq d$) a uma taxonomia onde a_{nov} é um conceito atômico não presente na KB e d um dado conceito. De seguida seguem-se os passos:

- Calcular, em primeiro lugar, S que é o mais específico dos subsumidores de d , isto é, o conjunto de conceitos atômicos d tais que $KB \models (d \sqsubseteq a)$ mas tal que não existe a' outro que não a tal que $KB \models (d \sqsubseteq a')$ e $KB \models (a' \sqsubseteq a)$;
- Calcular, em seguida, G que é o mais genérico dos subsumidos de d , isto é, o conjunto de conceitos atômicos a na taxonomia tal que $KB \models (a \sqsubseteq d)$ mas tal que não existe a' outro que não a tal que $KB \models (a' \sqsubseteq d)$ e $KB \models (a \sqsubseteq a')$;
- Se existe um conceito a em $S \cap G$ então o novo conceito a_{nov} está já presente da taxonomia com outro nome (nomeadamente a) e nenhuma ação é necessária;
- Caso contrário se existem ligações entre os conceitos em G e os conceitos em S estas são removidas pois a_{nov} irá ser colocado entre os dois grupos;
- De seguida são adicionadas as ligações, se existem, de a_{nov} a cada conceito em S e de cada conceito em G para a_{nov} ;
- Para tratar as constantes: é calculado C , o conjunto de constantes c na taxonomia tal que para cada $a \in S$, $KB \models (c \rightarrow a)$ mas tal que não inclua um $a' \in G$ tal que $KB \models (c \rightarrow a')$. (É feito por interseções definindo as diferenças nos conjuntos de constantes abaixo dos conceitos.) Depois para cada $c \in C$ testamos se $KB \models (c \rightarrow d)$ e se sim removemos as ligações de c para os conceitos em S e adicionamos uma única ligação de c para a_{nov} .

Para adicionar uma frase ($a_{\text{nov}} \sqsubseteq d$) a uma taxonomia é seguido um processo semelhante, mas mais simples. Como a_{nov} é uma nova primitiva não irão existir conceitos ou constante abaixo na taxonomia. Então só é necessário ligar a_{nov} aos subsumidores mais específicos de d . Para calcular estes começamos no topo da taxonomia com o conjunto [*Thing*] como primeiro S . Considerando S uma lista de subsumidores de d , supondo que um $a \in S$ tem pelo menos um descendente a' imediatamente abaixo dele na taxonomia tal que $KB \models (d \sqsubseteq a')$. De seguida remove-se a de S e substitue-se com os descendentes a' . O processo continua até nenhum elemento de S tenha um descendente que subsuma d . Se um conceito atômico a' abaixo de $a \in S$ que não subsume s então não haverá nenhum outro conceito abaixo desse a' durante a classificação. De notar que é possível ignorar uma árvore de descendentes completa, de forma segura, se a' for o suficientemente elevado na taxonomia. É o fulcro que permite classificar eficientemente mesmo para KB de grande dimensão. Para calcular os subsumidos mais genéricos G de um conceito d começamos pelos mais específicos S do primeiro G . Como d é subsumido pelos elementos de S sabemos que cada conceito abaixo de d estará, também, abaixo de S . Novamente outras partes da taxonomia mais distantes não serão utilizadas. Supondo um $a \in G$ e não é o caso $KB \models (a \sqsubseteq d)$ então remove-se a de G e substitui-se por todos os descendentes de a (ou removê-lo por inteiro se não os tem). O processo é repetido até

todos os elementos de G fiquem subsumidos por d . Depois apagamos todos os $a \in G$ que o ascendente também é subsumido por d .

Se a construção da taxonomia desta forma esta KB estará em posição de permitir a busca de forma mais eficiente. Para descobrir todas as constantes que satisfazem a questão q classificamos q e depois recolhemos todas as constantes no limite da árvore abaixo q . O que envolve um percurso na árvore na única parte da taxonomia subentendida por q . Similarmente para encontrar todos os conceitos atômicos que são satisfeitos pela constante c , iniciamos o percurso em c e subimos pela árvore recolhendo todos os conceitos que podem ser atingidos pelas ligações que representam a subsunção. As taxonomias derivadas pela descrição na KB da descrição lógica parecem-se com as hierarquias das *frames*. Nesse caso a hierarquia poderia ser criada da forma que se desejasse adicionando os *fillers* nos *slots* que conforme o apropriado. Contudo na descrição lógica é necessário lidar com o significado dos conceitos bem como os seus ascendentes e descendentes na taxonomia. Pelo que não é possível criar uma hierarquia ou mudança arbitrária na taxonomia, sendo necessário respeitar as relações implícitas nas estruturas dos conceitos. Na noção de herança as *frames* individuais tinham os valores, e procedimentos representados mais acima na hierarquia. O mesmo ocorre na taxonomia descrita: uma constante na taxonomia é considerada como tendo todas as propriedades que aparecem localmente (parte direita da frase onde foi introduzida) e em qualquer ascendente da taxonomia. A herança é mais simples porque é *estrita*, não existem exceções permitidas pela lógica dos operadores de formação de conceito. A propagação das propriedades a um indivíduo pode provocar a reclassificação de outras constantes, ao adicionar uma frase ($c \rightarrow d$) à KB mesmo que c já estivesse classificado.

7.6. HERANÇA

Quando consideramos o trabalho realizado relativamente a objetos pensamos em termos de hierarquias ou taxonomias o que reflete a importância da abstração, classificação e generalização no processo de representação do conhecimento. Os grupos de elementos partilham características comuns e falamos de determinados elementos de forma mais concisa utilizando palavras abstratas que representam estes conceitos. Permitem eliminar a redundância de representação, se afirmarmos que um animal é um mamífero ficamos a saber muita informação sobre o animal. A taxonomia dos tipos de objetos está presente em todo o lado especialmente quando organizamos conhecimento de forma compreensível. O raciocínio através da taxonomia permite concluir que um elemento herda as propriedades da sua generalização, no caso de um animal que é um mamífero concluímos que deve partilhar as características dos animais que, também, são mamíferos. Através das *frames* e das descrições anteriores a herança servia para realizar um certo tipo de raciocínio lógico de uma de uma forma graficamente orientada. Assim podemos pensar numa rede em que estes conceitos de especialização surgem abaixo das suas generalizações permitindo a herança das propriedades nas suas instâncias.

As *frames* e descrições abordadas anteriormente são igualadas a nós que aparecem *em redes de heranças*. Os conceitos de tipos e de atributos são, também considerados, como nós – Figura 7.4. Os seguintes conceitos são utilizados na discussão:

- *Aresta*: conecta dois nós diretamente representando relações de instância ou generalização;
- *Caminho*: este é constituído por a sequência de um ou mais nós;
- *Conclusão*: é suportada pelos caminhos, as arestas representam relações que são transitivas.

A forma mais simples de herança é baseada na descrição lógica e outros sistemas baseados na lógica clássica: *herança estrita*. Neste caso as conclusões são produzidas pelo completo encerramento de todos os caminhos da rede. Numa estrutura em forma de árvore e herança será simples. Todos os nós alcançáveis a partir de um determinado nó são implicados numa herança estrita que é um gráfico direto acíclico – *directed acyclic graph* (DAG) – os resultados são os mesmo que para as árvores: todas as conclusões que se podem alcançar por qualquer caminho são suportadas. As redes apresentam uma polaridade – positiva ou negativa – que se representa com uma barra sobre o caminho. A herança neste tipo de redes é, geralmente, designada de *herança múltipla* quando um nó tem mais de que um nó ascendente: nestes casos, pelo significado de aresta, o nó herda de todos os seus ascendentes. No sistema de *frames* pode haver casos de herança não estrita. Nessa representação as propriedades nem sempre se aplicam: nesse caso podem ser *derrotadas* ou sobrescritas. Sendo o exemplo mais óbvio o caso dos valores que os *slots* herdam, mas que na especialização podem ser sobrescritos.



Figura 7.4 – Rede de herança simples [45]

Na lógica desses sistemas virtualmente todas as propriedades, e procedimentos, podem ser sobrescritos. Este caso de herança designa-se por *herança derrotável*. O que complica o raciocínio derrotável é o facto que é possível que diferentes caminhos numa rede possam suportar conclusões contraditórias. No exemplo mostrado, Figura 7.5, é simples chegar a uma conclusão devido à asserção feita pela aresta negativa.



Figura 7.5 – Herança derrotável [45]

É simples verificar que vários casos poderão aparecer que não é possível escolher a melhor conclusão tornando-se ambígua. Para resolver estes casos diversas estratégias foram desenvolvidas. Uma é a designada por *Heurística de Caminho mais Curto*. A decisão automática através desta heurística deve preferir as conclusões resultando dos caminhos mais curtos da rede. A intuição por detrás desta escolha é que o caminho mais curto faz a herança de classes subsumidas mais específicas. De outra forma entre a escolha de superclasses para propriedade escolhemos o valor da mais específica, pois é espectável que seja a mais relevante. De notar que nem todos os caminhos contam para a geração de conclusões. Faz sentido considerar os caminhos como argumentos no suporte das conclusões. Alguns são *impedidos* por outros. Os que não o são podem ser designados por *admissíveis*. Assim o processo é decidir que conclusões admissíveis são suportadas pela rede. Apesar de plausível a heurística tem falhas. Quando a rede tem arestas redundantes – aquelas que já são implicadas pela rede – pode produzir conclusões erradas. Um outro problema é que a dimensão do caminho não reflete necessariamente o problema. Dependendo do problema alguns caminhos podem representar uma herança minuciosa enquanto outros uma herança menos minuciosa e mais geral. O facto de o caminho ser maior não deverá ser um óbice para retirar conclusões.

A heurística é considerada como uma estratégia *preemptiva* que permite escolher hipóteses admissíveis entre vários caminhos. Tenta produzir um critério de especificidade para ancorar a nossa intuição que a informação mais específica sobre um elemento é mais relevante que a generalização sobre o conjunto de que faz parte. Uma outra estratégia é a *distância inferencial* que em vez de ser baseada na distância linear baseia-se na topologia. Um nó a é considerado mais perto do nó b do que o nó c se e só se existe um caminho de a a c através de b independentemente da dimensão de quaisquer caminhos de a para b e para c . Não é isento de problemas como, por exemplo, quando o caminho a para c através de b é, por sua vez, contradito por um outro caminho. Lynn Stein apresentou uma formalização das redes de herança, que se apresenta:

- Uma *hierarquia de herança* $\Gamma = \langle V, E \rangle$ é um grafo direto e acíclico com arestas positivas e negativas para denotar “(normalmente) *is-a*” e “(normalmente) *is-not-a*” respetivamente. V representam os nós, ou vértices, e E as arestas. As arestas positivas são representadas por (a, x) e as negativas por $(a, \neg x)$;
- Um *caminho positivo* é uma sequência de zero ou mais arestas positivas: $a. \dots x$;
- Um *caminho negativo* é uma sequência de zero ou mais arestas positivas seguidas por uma aresta negativa: $a. \dots v. \neg x$. Um caminho ou é positivo ou negativo;
- Notar que não existem caminhos com mais de que uma aresta negativa, embora um caminho pode não ter aresta positivas – ser só a aresta negativa;
- Um caminho (ou *argumento*) suporta uma *conclusão* das seguintes formas:
 - $a. \dots x$ suporta a conclusão $(a \rightarrow x)$ (a é um x);
 - $a. \dots v. \neg x$ suporta a conclusão $(a \nrightarrow x)$ (a não é um x).
- Uma conclusão pode ser suportada por vários argumentos. Contudo nem todos os argumentos são igualmente aceitáveis. A definição formal da *admissibilidade* providencia o mecanismo para que um argumento prevaleça:

- Γ *suporta um caminho* se o conjunto de arestas correspondentes estão em E e o caminho é admissível de acordo com a definição que se segue: A hierarquia suporta a conclusão $a \rightarrow x$ (ou $a \rightarrow x$) se suporta algum caminho correspondente entre a e x ;
- Um caminho $a.s_1 \dots s_n.(\neg)x$ é admissível se todas as arestas em si são admissíveis relativamente a a ;
- Uma aresta $v.(\neg)x$ é *admissível em Γ com respeito a a* se existe um caminho positivo $a.s_1 \dots s_n.v, (n > 0)$, em E e:
 - Cada aresta em $a.s_1 \dots s_n.v$ é admissível em Γ com respeito a a (recursivamente);
 - Nenhuma aresta em $a.s_1 \dots s_n.v$ é redundante em Γ com respeito a a ;
 - Nenhum nó intermediário a, s_1, \dots, s_n é *preemptivo* de $s_1.(\neg)x$ em respeito a a .
- Então uma aresta é admissível com respeito a a se existe um caminho, não redundante, admissível de a até ao destino que não contenha intermediários impeditivos;

A definição de *preemptivo* ao longo do caminho e de *redundância* completa a formalização básica:

- Um nó ao longo de um caminho $a. \dots y \dots v$ é *preemptivo* de $v.x(v. \neg x)$ com respeito a a se $y. \neg x \in E(y.x \in E)$;
- Uma aresta positiva $b.w$ é *redundante* em Γ com respeito ao nó a se existe algum caminho positivo $b.t_1 \dots t_m.w \in E, (m \geq 1)$ em que:
 - Cada aresta em $b.t_1 \dots t_m$ é admissível em Γ com respeito a a (nenhuma das arestas em si mesmas são preemptivas);
 - Não existem c e i tais que $c. \neg t_i$ que é admissível em Γ a respeito de a ;
 - Não existe c tal que $c. \neg w$ que é admissível em Γ a respeito de a .

Depois da admissibilidade e preemptividade podemos ver quais as conclusões a ser aceites numa rede de herança. Não se espera que uma rede ambígua especifique um conjunto único de conclusões. Utilizamos o termo *extensão* para significar um conjunto de crenças suportadas pela rede. Formalmente teremos o seguinte:

- Γ é *a – conectada* se e só se para cada nó x em Γ existe um caminho de a para x e para cada aresta $v. \neg x$ em Γ existe um caminho positivo de a a v . Ou seja, cada nó e aresta é acedível a partir de a ;
- Γ é (potencialmente) ambígua com respeito ao nó a em x se existe um nó $x \in V$ tal que ambos $a.s_1 \dots s_n.x$ e $a.t_1 \dots t_m. \neg x$ são caminhos;
- Uma *extensão crédula* de uma hierarquia de herança Γ com respeito ao nó a é a máxima não-ambígua sub-hierarquia *a – conectada* de Γ com respeito a a .

Assim se X é uma extensão crédula de Γ então ao adicionar uma aresta de Γ a X faz com que X seja ou ambígua ou não *a – conectada*. Estas extensões não incorporam as noções de admissibilidade ou de preemptividade. Mas dada a relação do raciocínio com o mundo real que pretendemos podemos formalizar *extensões preferíveis*:

- Sejam X e Y extensões crédulas de Γ a respeito dum nó a . X é preferível a Y se e só se existem nós v e x tais que:
 - X e Y concordam em todas as arestas cujos destinos finais precedem v topologicamente;
 - Existe uma aresta $v.x$ (ou $v. \neg x$) que é *inadmissível* em Γ ; e
 - Esta aresta está em Y mas não em X .

Uma extensão crédula é uma *extensão preferível* se não existe uma outra extensão crédula preferível a si. No fundo um agente terá de decidir em qual representação acreditar. As extensões descritas oferecem um conjunto de conclusões consistentes, mas a atitude do agente pode variar. Foram propostas várias formas de raciocínio baseadas nas formalizações apresentadas:

- *Raciocínio crédulo*: escolher uma extensão preferível, talvez arbitrariamente, e acreditar em todas as conclusões suportadas;
- *Raciocínio cético*: acreditar nas conclusões suportadas por qualquer caminho que está presente em todas as extensões preferíveis;
- *Raciocínio cético ideal*: acreditar nas conclusões que são suportadas por todas as extensões preferíveis. Este é um pouco mais subtil em que as conclusões podem ser suportadas por caminhos distintos em cada extensão: o que prova a sua incapacidade de ser computado baseado no caminho.

Nestes processos observamos o raciocínio na direção dos ascendentes. Uma alternativa seria começar no ascendente principal e descer na árvore para observar a propagação na direção dos descendentes. [45]

7.7. INDEFINIÇÃO, INCERTEZA E GRAUS DE CRENÇA

Até agora foram apresentadas frases lógicas e precisas sobre o mundo de diferentes formas para representar o conhecimento. Mas quando tentamos emular o conhecimento do senso comum e raciocínio efetuado naturalmente a precisão lógica pode ficar curta. A sua definição é discutida há muito, desde os filósofos Gregos, e continua a ser um tópico não totalmente desmistificado. Existem demasiados exemplos a considerar: nem todos os pássaros voam. Ou seja, nem todos os pássaros apresentam as características que consideramos normais para um pássaro.

Por vezes não é apropriado a expressão do conhecimento com a totalidade de uma lógica universal. Nem tudo é exatamente algo, mas, por vezes mais alguma coisa. Nem sempre é possível medir ou quantificar algo de forma precisa, e por vezes nem é necessário. Nestes casos a utilização de informação limitada e regras imperfeitas podem permitir conclusões seguindo algumas premissas, mas não da forma abordada até agora. Apresentamos, muito resumidamente, algumas formas de expressar o conhecimento nestes campos.

7.7.1. RACIOCÍNIO NÃO BOOLEANO

A primeira ideia será a utilização da probabilidade. Esta expressa a possibilidade de algo acontecer ou que seja verdadeiro ou falso. A introdução de números, especialmente de números inteiros, parece permitir contornar a categorização natural dos valores lógicos binários. Com introdução da noção de “*menos de 100%*” na representação é possível ir de factos que eram tidos como irrefutáveis para alguma percentagem desses mesmos factos serem irrefutáveis. Mas nem esta opção permite abarcar todo o que se pretende representar. As probabilidades servem bem numa cadeia de eventos em que se pretende saber a probabilidade do próximo evento ocorrer, mas nem todo o nosso raciocínio e “palpites” se enquadram nesta forma de expressão. Quando queremos representar opiniões baseadas

em evidências e inferências sobre a probabilidade de um evento individual acontecer ou de uma característica, como um pássaro voar ou não, ou ainda de algum “alguém ser bastante rápido” as probabilidades não parecem ser o mais correto. Existem três tipos de modificação de forma a flexibilizar a estrutura da visão lógica sobre algo:

- Podemos afrouxar a força do *quantificador*. Invés de para *todo o x* podemos dizer para a *maior parte de x* ou *para 95% de x*. O resultado é uma asserção sobre a frequência, ou seja, uma interpretação *estatística*. A utilização da probabilidade neste caso é *objetiva* pois é sobre o mundo, puro e simples, em questões que não estão sujeitas a interpretações ou graus de confiança.
- Podemos afrouxar a força do *predicado*. Invés de asserções estritas, “Todos são ...”, podemos ter “Todos são *moderadamente* ...” sendo a propriedade em questão qualquer – altos, baixos, rápidos ou outros. Este predicado aplicado a um indivíduo numa maior ou menor extensão é um *predicado vago*. De notar que, assim, uma pessoa poderá ter simultaneamente duas características contrárias que variam de acordo com o peso que se dá cada uma.
- Podemos afrouxar o grau de crença na *frase*. Invés da frase “Todos são...” posso dizer “*Acredito* que todos são..., mas não tenho a certeza.”. Esta afirmação pode ter várias fontes, mas não reflete uma probabilidade nem um predicado categórico. É *conhecimento incerto*; quando se quantifica a falta de certeza utilizamos a noção de *probabilidade subjetiva* porque reflete um grau de crença e não a objetiva frequência de um evento.

Esta separação permite o seu estudo para a representação, mas a utilização em conjunto não é impossível.

7.7.2. PROBABILIDADE OBJETIVA

As probabilidades objetivas ocupam-se da *frequência*. Falamos de percentagem de acontecer algo, a longo prazo, quando as condições são as mesmas. A *possibilidade x* é a percentagem de vezes que *x* é espetável de acontecer numa sequência de vários eventos, em que o processo básico é repetido e cada evento é independente dos que já aconteceram e as condições são as mesmas. Este tipo de probabilidade que se ocupa de frequências fatuais é chamada *objetiva* porque não depende de quem verifica as probabilidades. Como é uma visão estatística não suporta diretamente a assigação de uma crença sobre um evento aleatório que não é parte de nenhuma sequência repetível óbvia. Tecnicamente uma probabilidade é um número entre 0 e 1, inclusive, representando a frequência de um evento num espaço, suficientemente largo, de amostras aleatórias. Um evento com probabilidade 1 é considerado acontecer sempre e com probabilidade 0 nunca acontece.

Formalmente começamos com um conjunto universal U de todas as ocorrências. Um evento a é um subconjunto de U . A *medida de probabilidade* \mathbf{Pr} é uma função dos eventos para os números no intervalo $[0, 1]$ satisfazendo os dois seguintes postulados básicos:

- $\mathbf{Pr}(U) = 1$;
- Se a_1, a_2, \dots, a_n são eventos disjuntos então $\mathbf{Pr}(a_1 \cup a_2 \cup \dots \cup a_n) = \mathbf{Pr}(a_1) \cup \dots \cup \mathbf{Pr}(a_n)$.

Destes vem $\mathbf{Pr}(\bar{a}) = 1 - \mathbf{Pr}(a)$ logo $\mathbf{Pr}(\{\}) = 0$. Também, considerando dois eventos a e b , temos que $\mathbf{Pr}(a \cup b) = \mathbf{Pr}(a) + \mathbf{Pr}(b) - \mathbf{Pr}(a \cap b)$. Outra consequência útil é a seguinte: se

b_1, b_2, \dots, b_n são eventos disjuntos e esgotam todas as possibilidades, isto é, se $(b_i \cap b_j) = \{\}$ para $i \neq j$, e $(b_1 \cup \dots \cup b_n) = U$ temos $\Pr(a) = \Pr(a \cap b_1) + \dots + \Pr(a \cap b_n)$.

Em caso de probabilidade é útil pensar numa simples interpretação de \Pr : sendo U um conjunto finito e $\Pr(a)$ é um número de elementos em a divididos pela dimensão de U , ou seja, a proporção de elementos de U que também estão em a . Esta interpretação satisfaz os postulados anteriores. A ideia chave na teoria de probabilidade é a *condicionalidade*. A probabilidade de um evento pode depender da sua interação com outros. Esta probabilidade *condicional*, $\Pr(a|b)$, significa a probabilidade de a acontecer devido a que b ocorreu. Na nossa interpretação $\Pr(a|b)$ significa a proporção dos elementos que estão em a entre os elementos de b . Formalmente:

- $\Pr(a|b) = \Pr(a \cap b)/\Pr(b)$, quando $\Pr(b)$ é zero a probabilidade é considerada indefinida.

Geralmente não é possível prever o valor de $\Pr(a \cap b)$ dados os valores de $\Pr(a)$ e $\Pr(b)$. Na nossa interpretação não é possível prever a dimensão de $(a \cap b)$ dadas apenas as dimensões de a e b . Da definição vem $\Pr(a \cap b) = \Pr(a|b) \times \Pr(b)$, e mais genericamente a *regra de encadeamento* $\Pr(a_1 \cap \dots \cap a_n) = \Pr(a_1|a_2 \cap \dots \cap a_n) \times \Pr(a_2|a_3 \cap \dots \cap a_n) \times \dots \times \Pr(a_{n-1}|a_n) \times \Pr(a_n)$. Também vem $\Pr(\bar{a}|b) = 1 - \Pr(a|b)$ e temos que se b_1, b_2, \dots, b_n são eventos disjuntos e esgotam todas as possibilidades temos $\Pr(a|c) = \Pr(a \cap b_1|c) + \dots + \Pr(a \cap b_n|c)$. A regra de *Bayes* utiliza a definição da probabilidade condicional para relacionar a probabilidade de um a dado b e a probabilidade de um b dado a : $\Pr(a|b) = \frac{\Pr(a) \times \Pr(b|a)}{\Pr(b)}$

Um evento a é condicionalmente independente de b se b não afeta a probabilidade de a , ou seja, $\Pr(a|b) = \Pr(a)$. Na interpretação a é independente de b se a proporção de elementos de a no conjunto b é igual à proporção de a na população geral U . Da definição vem que um evento a é independente de b se e só se $\Pr(a \cap b) = \Pr(a) \times \Pr(b)$, se e só se b é independente de a . Logo a relação de *independência condicional* é simétrica. Também a e b são *condicionalmente independentes dado c* se $\Pr(a|b \cap c) = \Pr(a|c)$.

Para determinar possibilidade de um acontecimento a dado tudo o que é conhecido não é possível sabendo só algumas probabilidades condicionais de a . Saber o valor de $\Pr(a|c)$ não é suficiente quando queremos $\Pr(a|b \cap c)$ porque a probabilidade de um a dados ambos b e c não é relacionada com a probabilidade de a dado só c , a não ser que a seja independente de b dado c .

7.7.3. PROBABILIDADE SUBJETIVA

A *subjetividade* do grau de confiança ou certeza numa frase por um agente é separável do conteúdo proposicional da frase em si. Estes graus de confiança, ou crença advêm, muitas vezes da observação, sobre grupos de elementos no mundo descrito. Estes graus de crença geralmente derivam de considerações estatísticas sendo usualmente referidos como *probabilidades subjetivas*. Geralmente são

trabalhadas vendo como a combinação de evidências e modificam a crença sobre o mundo em vez de retirar novas conclusões. São definidos dois tipos de probabilidades relativamente ao processo de obter conclusões:

- Probabilidade *a priori* de uma frase α envolve o anterior estado da informação ou conhecimento prévio designado por β : $\Pr(\alpha|\beta)$;
- Probabilidade *a posteriori* é derivada quando nova evidência, γ , é apresentada: $\Pr(\alpha|\beta \wedge \gamma)$.

Se numa população existe a probabilidade de 2% de uma determinada característica, o grau de crença que um indivíduo aleatório terá de ter essa mesma característica é .002. Esta é a probabilidade subjetiva *a priori*. Se observarmos o indivíduo em questão podemos adicionar outras evidências que dada a probabilidade *a priori* e as estas novas evidências modificam o grau de crença para .65. Sendo assim o ponto chave será como combinar as evidências de várias fontes para reavaliação das crenças.

Para passar das estatísticas para as crenças podemos, intuitivamente, derivar as crenças das estatísticas. Este processo usa a abordagem tradicional de abordar uma *classe de referência*, para a qual temos informação estatística, e utilizar a mesma para computar um grau apropriado de crença para o indivíduo. A classe de referência seria uma classe geral onde o indivíduo em questão pareceria ser confortavelmente associado. Esta simples e direta inferência trás vários problemas por causa das múltiplas classes de referência. Uma abordagem *Bayesiana* básica permite assumir um número de variáveis proposicionais (ou frase atómicas) de interesse, p_1, p_2, \dots, p_n . Nos diferentes estados do mundo diferentes combinações destas variáveis serão verdade. Podemos associar cada estado do mundo a uma caracterização através de uma interpretação \mathcal{J} que especifica quais as frases atómicas que são verdadeiras e quais são falsas. A uma *probabilidade de distribuição conjunta* J admitimos uma especificação do grau de crença para cada um das 2^n afirmações de verdade para as variáveis de proposicionais. Ou seja, para a interpretação \mathcal{J} , $J(\mathcal{J})$ é um número entre 0 e 1 tal que $\sum J(\mathcal{J}) = 1$, sendo a soma sobre as 2^n possibilidades. Este cenário apresenta o agente como não sabendo o verdadeiro estado do mundo e $J(\mathcal{J})$ é o grau de crença que o agente assigna ao estado mundo especificado por \mathcal{J} . Utilizando esta probabilidade é possível calcular o grau de crença de cada frase envolvendo qualquer subconjunto das variáveis. O grau de crença em α é a soma de J sobre todas as interpretações onde α é verdadeiro. Ou seja, acreditamos em α como acreditamos nos estados que satisfazem α . Formalmente definimos $\Pr(\alpha) \stackrel{def}{=} \sum_{\mathcal{J}=\alpha} J(\mathcal{J})$ onde $\Pr(\alpha|\beta) = \Pr(\alpha \wedge \beta) \div \Pr(\beta)$. Embora a abordagem apresente resultados e nos permita calcular qualquer probabilidade subjetiva assume uma distribuição de probabilidade conjunta sobre todas as variáveis que nos interessam. Isto implica que para n frases atómicas seria necessário especificar os valores de $2^n - 1$ números, que seria impraticável. Para limitar o que é necessário ter em conta para o raciocínio sobre as probabilidades subjetivas é necessário utilizar algumas suposições simplificativas. Assumindo frases atómicas p_1, p_2, \dots, p_n podemos especificar uma interpretação (P_1, P_2, \dots, P_n) onde cada P_i é ou p_i (quando a frase é verdadeira) ou $\neg p_i$ (quando é falsa). Da definição observamos que

$J(\langle p_1, p_2, \dots, p_n \rangle) = \Pr(P_1 \wedge P_2 \wedge \dots \wedge P_n)$, porque existe uma única interpretação que satisfaz a conjunção de todos os literais. Uma outra simplificação extrema seria assumir que todas as frases atômicas seriam condicionalmente independentes entre si. Ou seja, $J(\langle p_1, p_2, \dots, p_n \rangle) = \Pr(P_1) \cdot \Pr(P_2) \cdot \dots \cdot \Pr(P_n)$. Nesta simplificação apenas seriam necessários calcular n números para satisfazer a probabilidade conjunta de distribuição. Este cálculo é demasiado extremo pois existem sempre dependências entre as frases atômicas.

Chamamos *rede de crenças* (ou rede *Bayesiana*) à representação de todas as variáveis p_i num gráfico direto acíclico. Intuitivamente deverá existir uma ligação de p_i para p_j se pensarmos que a verdade do primeiro afeta a verdade do segundo, neste caso p_i é o ascendente de p_j na rede de crenças. Supondo que numeramos as variáveis de forma que os ascendentes de cada variável p_j aparecem anteriormente do que p_j , o grafo é acíclico. Segundo a regra de encadeamento dá-se o caso $J(\langle p_1, p_2, \dots, p_n \rangle) = \Pr(P_1) \cdot \Pr(P_2|P_1) \cdot \Pr(P_3|P_1 \wedge P_2) \cdot \dots \cdot \Pr(P_n|P_1 \wedge \dots \wedge P_{n-1})$. Formulado desta forma continua a ser necessário $2^n - 1$ números porque para cada termo existem 2^j probabilidades condicionais para especificar. Contudo vamos assumir que na rede de crenças cada variável proposicional é condicionalmente independente das variáveis ascendentes que não são suas progenitoras dadas as suas progenitoras. Ou seja, $\Pr(P_{j+1}|P_1 \wedge \dots \wedge P_j) = \Pr(P_j|\text{progenitores}(P_{j+1}))$ onde $\text{progenitores}(P_{j+1})$ é a conjunção daqueles literais P_1, P_2, \dots, P_j são progenitores de P_{j+1} no grafo. Considerando estas assunções vem que $J(\langle p_1, p_2, \dots, p_n \rangle) = \Pr(P_1|\text{progenitores}(P_1)) \cdot \Pr(P_2|\text{progenitores}(P_2)) \cdot \dots \cdot \Pr(P_n|\text{progenitores}(P_n))$.

A utilização desta equação define a probabilidade de distribuição conjunta J da probabilidade que interessa calcular. Para especificar J é necessário saber para cada variável p . Se p tem k progenitores na rede é necessário saber 2^k probabilidades condicionais, correspondente às possibilidades de verdade ou falsidade de cada progenitor. Somando sobre todas as variáveis teremos, no máximo, $n2^k$ números a especificar, onde k é o máximo de progenitores para qualquer nó. Com n a crescer o número esperado será muito menos que 2^n . A utilização da probabilidade de distribuição conjunta para computar o grau de crença e como utilizar o conjunto de todas as interpretações lógicas para computar as implicações. Faz o correto, mas apenas para conjuntos de dimensão reduzida. Calcular o grau de crença de rede de crença pode ser NP-difícil, mas o desenvolvimento de processos de raciocínio especializados aparentam funcionar bem em problema práticos ou redes com topologias de árvores restritas.

Estas redes são úteis para computar as probabilidades subjetivas baseadas em assunções independentes e relações causais. Mas decisões com incerteza implicam, usualmente, outros fatores a ter em conta, como o mérito relativo de diferentes resultados e os seus custos. Os diagramas de influência tentam estender o raciocínio das redes de crenças com maior conjunto de nós tipo. São

considerados quatro tipos de nós: *nós de oportunidade* que representam as variáveis probabilísticas, *nós determinísticos* que representam as computações baseadas nas suas entradas, *nós de decisão* que representam as decisões tudo ou nada feitas pelo utilizador e o *nó valor*, único, que representa a decisão final baseada em alguma função de avaliação. As ligações entre representam as relações entre nós (probabilística e determinísticas). A intenção deste tipo de diagramas é para um sistema raciocinar sobre as relações ente variáveis que são probabilisticamente determinadas, determinadas por escolha e deterministicamente determinadas. Permite uma plataforma para suportar um sistema de decisão, e um número de sistemas implementadas para raciocinar com estas representações.

7.7.4. INCERTEZA

Para além dos processos descritos anteriormente podemos considerar o grau que cada predicado é satisfeito. Consideramos uma determinada característica e fazemos uma afirmação sobre a mesma. Essa afirmação tem um grau de incerteza pois será possível fazer comparações dessa característica. Estes predicados intuitivamente pensados têm um certo grau são chamados de *predicados vagos*. Correspondem, mais ou menos, a adjetivos que podem ser modificados através de advérbios. Consideramos para cada predicado uma *função base* precisa em termos que o predicado é entendido. Podemos capturar a relação entre o predicado vago e a sua função base através de uma *curva de grau*, Figura 7.6. Nesta podem ser considerados vários predicados em que o grau de cada característica se aplica ao elemento. Ou seja, o grau de satisfação pode ser não zero para múltiplos predicados sobre a mesma função base, mesmo os que aparentemente são antagónicos.

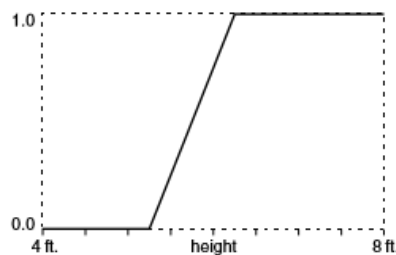


Figura 7.6 – Curva de grau de um predicado vago [45]

Como com acontece com a lógica e probabilidade é necessário considerar a combinações booleanas das propriedades vagas e ver que graus estas são satisfeitas. A negação é simples, vem $\mathbf{Pr}(\neg p) = 1 - \mathbf{Pr}(p)$. As conjunções e disjunções parecem ter características especiais. Imaginando que se necessita de um candidato com certas características, com grau de satisfação de .95 para as características. Um candidato que satisfizesse todas com este grau deveria ser ideal, mas, considerando vinte características, teríamos como grau de satisfação conjunta apenas $(.95)^{20} = .36$ o que não parece ser o que se procura. Então existe uma diferença entre a *probabilidade* de satisfazer o critério conjunto, considerando a independência, seria o produto das probabilidades de satisfazer cada critério individual, e o *grau* que cada critério é satisfeito. O grau que um indivíduo é P e Q é o *mínimo* dos graus que é P e é Q . De forma similar o grau que o critério *disjunto* é satisfeito é o *máximo* grau que

cada critério individual é satisfeito. Uma aplicação interessante dos predicados vagos é no chamado, por vezes, controlo difuso – *fuzzy control*. Neste processo temos um encadeamento de regras que aplicam os graus de satisfação anteriores para tomar decisões e controlar o processo. A vantagem é poder tomar decisões através de inferências mesmo quando as condições anteriores não são totalmente satisfeitas. Estas regras são geralmente desenvolvidas em grupo e não são tidas como significativamente independentes umas das outras; o seu principal objetivo é de trabalhar para conjuntamente afetar o resultado.

O procedimento anterior apresenta aplicabilidade em vários campos. Mas apresentam-se que apesar diferenças concetuais entre os graus de crença e os graus de satisfação muito do raciocínio com os predicados vagos podem ser realizados em termos de probabilidade subjetiva. Esta seria uma reconstrução *Bayesiana* em que a questão seria qual o grau de crença numa determinada característica dado um valor para a mesma. Se assumirmos $\Pr(\alpha \wedge \beta | \gamma) = \min\{\Pr(\alpha | \gamma), \Pr(\beta | \gamma)\}$ temos a resolução das combinações das propriedades. Desde que não assumamos que α e β são independentes. Além disso observamos que $\Pr(\alpha \vee \beta | \gamma) = \max\{\Pr(\alpha | \gamma), \Pr(\beta | \gamma)\}$ pelas propriedades gerais da probabilidade.

7.8. CONCLUSÃO

Neste capítulo discutimos a representação do conhecimento. Começamos por ver o que é conhecimento e como representá-lo de forma a poder raciocinar sobre o mesmo. Apresentamos algumas observações como o conhecimento tem de ser abordado para poder representá-lo pois é necessário passar do mundo real para um outro domínio que representa, sem perder a fiabilidade, esta informação. Como ferramenta utilizamos a FOL para representar e raciocinar sobre o mesmo. De seguida resumimos algumas formas de representar e raciocinar sobre o conhecimento. Estas apresentam características que são capazes de responder a diversas solicitações. Terminamos com uma forma de representar e raciocinar sobre conhecimento que não é exatamente a representação do conhecimento, mas faz observações através da probabilidade de algo. O que permite tomar decisões mesmo na presença de factos incompletos.

8. DINÂMICA DO CONHECIMENTO

Neste capítulo iremos apresentar um complemento do discutido até agora. Apresentamos algumas opções de como representar o conhecimento e raciocinar sobre o mesmo. A questão que se põe é como alterar o conhecimento já presente quando na presença de nova informação a integrar. Ou, mesmo, como decidir em presença de informação contraditória.

8.1. REVISÃO DE CRENÇAS

A *Revisão de Crenças* é um processo que define a modificação das crenças de um agente perante nova informação. Este tópico, iniciado nos anos 80, foi desenvolvido com contribuição de duas grandes áreas: a Ciência da Computação e Filosofia. Na primeira a necessidade de atualizar bases de dados de forma correta levou que os investigadores de IA construíssem modelos cada vez mais complexos para a sua manutenção. No campo da Filosofia os trabalhos de Levi [47] [48] e Harper [49] foram os pioneiros. Sendo o trabalho mais influente realizado por Carlos Alchourrón, Peter Gärdenfors e David Makinson em 1985 [50] o modelo AGM. Este apresenta uma base geral e versátil para o estudo de revisão de crenças. A partir desta base muitos desenvolvimentos aconteceram tais como operadores de mudança, aplicabilidade do modelo entre outras. Em [51] Fermé e Hanson apresentam um sumário dos 25 primeiros anos de estudos e desenvolvimentos do modelo AGM.

8.1.1. AGM

Em [52] o modelo AGM representa o estado como um conjunto logicamente fechado de frases (conjunto de crenças), mas numa abordagem alternativa pode ser considerado como não sendo fechado (crenças base). O modelo AGM apresenta as crenças como um conjunto de frases numa determinada linguagem formal. Sendo \mathcal{L} a linguagem as frases serão apresentadas por letras minúsculas (p, r, s, t, u, \dots) e o conjunto das frases como maiúsculas (W, X, Y, Z). As letras maiúsculas (K, K', M, M', \dots) representam o conjunto de frases fechado sob consequência lógica. Iremos assumir, também, que a linguagem contém os conectores lógicos habituais: negação (\neg), conjunção (\wedge), disjunção (\vee), implicação (\rightarrow) e equivalência (\leftrightarrow). Os símbolos (\perp) e (\top) denotam uma contradição e tautologia arbitrárias respetivamente.

As crenças de um agente são representadas pelo conjunto fechado sob consequência lógica. Para qualquer conjunto X $C_n(X)$ é o conjunto de consequências lógicas de X . Uma operação de consequência numa linguagem é uma função C_n de um conjunto para um conjunto de frases e satisfaz as seguintes condições:

- *Inclusão:* $X \subseteq C_n(X)$;
- *Monotonia:* se $X \subseteq Y$ então $C_n(X) \subseteq C_n(Y)$;
- *Iteração:* $C_n(X) = C_n(C_n(X))$;
- *Supraclássica:* se p pode ser derivado logicamente de X então $p \in C_n(X)$.

Afirmamos que $K \in L$ é um conjunto de crenças se e só se $K = C_n(K)$. $X \vdash p$ A notação $X \vdash p$ é uma alternativa para $p \in C_n(X)$. A notação $X \not\vdash p$ serve o mesmo propósito de $p \notin C_n(X)$. O conjunto de tautologias é representado por $C_n(\emptyset)$.

No modelo AGM existem três critérios de racionalidade:

- **Prioridade de nova informação:** a nova informação é sempre aceite;
- **Consistência:** se possível o novo estado deve ser consistente;
- **Economia informacional:** reter o mais possível as crenças preexistentes.

Os três tipos de mudança de crenças presentes na estrutura AGM são: contração, expansão e revisão. Na contração uma frase especificada p é removida. O que indica que o conjunto K foi substituído por um conjunto $K \div p$ que é um conjunto K que não contém p . Na expansão uma frase p é adicionada a K e nada é removido. Significa que o conjunto K é substituído por um conjunto $K + p$ que é o menor conjunto logicamente fechado que contém K e p . Caracteriza-se por $K + p = C_n(K \cup p)$. Na revisão uma frase p é adicionada a K e, ao mesmo tempo, uma ou outras frases são removidas para que o conjunto de crenças ($K * p$) seja consistente.

8.1.1.1. CONTRAÇÃO

Na contração de K por p o resultado deve ser um subconjunto de K que não implica p . Terá de ser consistente e incluir o máximo de subconjuntos de K que não implicam p . A contração de um conjunto de crenças deve satisfazer seis postulados básicos do AGM:

- *Fecho:* $K \div p = C_n(K \div p)$;
- *Sucesso:* se $p \notin C_n(\emptyset)$ então $p \notin C_n(K \div p)$;
- *Inclusão:* $K \div p \subseteq K$;
- *Vacuidade:* se $p \notin C_n(K)$ então $K \div p = K$;
- *Extensionalidade:* se $p \leftrightarrow q \in C_n(\emptyset)$ então $K \div p = K \div q$;
- *Recuperação:* $K \subseteq (K \div p) + p$.

O postulado de recuperação é um pouco frágil [51] pois indica que fica tanto depois da remoção de p de K que é possível voltar ao estado anterior pela expansão de K por p . Em algumas ocasiões este postulado não é satisfeito – bases de crenças. Para caracterizar contrações não atômicas – conjunções – mais dois postulados suplementares devem ser considerados:

- *Inclusão conjuntiva:* se $p \notin K \div (p \wedge q)$ então $K \div (p \wedge q) \subseteq K \div p$;
- *Sobreposição conjuntiva:* $(K \div p) \cap (K \div q) \subseteq K \div (p \wedge q)$.

Os primeiros seis postulados caracterizam a contração parcial e os oitos a contração parcial relacional.

8.1.1.2. REVISÃO

O operador de revisão inclui duas tarefas de grande importância: a de adicionar a nova crença p ao conjunto de crenças K e garantir que o conjunto $K * p$ resultante é consistente (excluindo o caso de p ser inconsistente).

A tarefa inicial será conseguida utilizando a expansão por p . A segunda através da contração por $\neg p$, ou seja, um conjunto de crenças que não implica $\neg p$ para que p possa ser adicionado sem perda de consistência. Denominada por *Identidade de Levi*.

O operador $*$ é um operador de revisão se e só se satisfaz os seguintes postulados:

- *Fecho*: $K * p = C_n(K * p)$;
- *Sucesso*: $p \in K * p$;
- *Inclusão*: $K * p \subseteq K + p$;
- *Vacuidade*: se $\neg p \notin K$ então $K * p = K + p$;
- *Consistência*: $K * p$ é consistente se p é consistente;
- *Extensionalidade*: se $p \leftrightarrow q \in C_n(\emptyset)$ então $K * p = K * q$.

Adicionalmente os dois seguintes postulados fazem parte do conjunto:

- *Superexpansão*: $K * (p \wedge q) \subseteq (K * p) + q$;
- *Subexpansão*: se $\neg p \notin C_n(K * p)$ então $(K * p) + q \subseteq K * (p \wedge q)$.

Os primeiros seis postulados caracterizam a revisão parcial e os oitos a revisão parcial relacional transitiva.

8.1.1.3. IDENTIDADES DE HARPER E LEVI

As funções de contração e revisão são caracterizadas por dois, diferentes, conjuntos de postulados. Estes são independentes, e cada um não refere os postulados do outro. Contudo a revisão pode ser definida através da contração e vice-versa [53]. São as *Identidade de Harper* e *Identidade de Levi*. Através da Identidade de Levi é possível definir a revisão em termos de contração: $K * p = (K * \neg p) + p$.

A revisão consiste de duas operações:

- Contração de K por $\neg p$ e obtenção, se possível, de um subconjunto K consistente com p ;
- Expansão do resultado por p .

A identidade de Harper permite definir a contração em termos de revisão: $K \div p = K \cap (K * \neg p)$.

A contração é definida pela interseção de K com a revisão de K por $\neg p$. Como discutido a revisão e contração podem ser definidas e a outra pode ser univocamente definida.

8.1.1.4. CONTRAÇÃO SEGURA E DE NÚCLEO

A operação proposta por Alchourrón e Makinson [54] de *Contração Segura* é baseada numa relação, $<$, não circular nos elementos de K . Um elemento a de K é seguro com respeito a p se e só se todos os subconjuntos mínimos de inclusão de K que impliquem p não contenham a ou contenham um qualquer b tal que $b < a$. A contração segura baseada em $<$ fornece um resultado lógico de fecho no conjunto de frases em K que são seguras com respeito a p . Esta operação é uma contração parcial. Hanson [28] introduziu a *Contração de Núcleo*. Esta é uma generalização não-relacional da contração segura. Sendo Kp o conjunto mínimo de subconjunto de K que implicam p , a função σ de incisão seleciona frases a serem descartadas. Esta satisfaz as seguintes propriedades:

- $\sigma(K \perp p) \subseteq \cup (K \perp p)$;
- Se $\emptyset = X \in (K \perp p)$ então $X \cap \sigma(K \perp p) \neq \emptyset$.

A contração do núcleo \approx_σ baseada em σ é definida pela relação $K \approx_\sigma p = K \setminus \sigma(K \perp p)$. A contração do núcleo sustenta os seis postulados básicos AGM e satisfaz a condição adicional de suavidade, nomeadamente que se $X \subseteq K$ e $C_n(X) \cap \sigma(K \perp p) \neq \emptyset$ então $X \cap \sigma(K \perp p) \neq \emptyset$.

8.1.2. BASE DE CRENÇAS

Um conjunto de crenças é fechado sob consequência lógica. Mas existem crenças que podem não ser consideradas [52]. Tomando um exemplo clássico, supondo um conjunto de crenças contém a frase p – Shakespeare escreveu a peça Hamlet – devido ao fecho lógico deve também conter a frase $p \vee q$ – Ou Shakespeare escreveu a peça Hamlet ou Charles Dickens escreveu a peça Hamlet – que é uma consequência lógica que não deveria ter suporte por si só. Um conjunto de frases que não é fechado sob consequência lógica é uma Base de Crenças. Como dito em [52] os elementos deste conjunto representam crenças que são mantidas independente de outra qualquer crença ou conjunto de crenças. As mudanças nesta base deverão ter em atenção que, pela intuição, as crenças que são meramente derivadas não são relevantes de manter só por si. Quando uma que as suporta deixa de ser mantida, esta, também, deverá ser descartada. Em cada base de crenças X existe um conjunto $C_n(X)$ que representa as crenças suportadas de acordo com X . Por outro lado, um conjunto pode ser representado por diferentes bases. As vantagens podem ser resumidas:

- As bases de crenças apresentam um maior poder de expressividade relativamente aos conjuntos de crenças. Poderão sere estatisticamente equivalentes, representam as mesmas crenças, mas podem não ser dinamicamente equivalentes no seu comportamento sob operações de mudança. Ou seja, podem representar diferentes formas de conter as mesmas crenças;
- As bases permitem representar agentes limitados em termos de recursos. Um agente ideal terá recursos infinitos. Contudo um agente real não os tem e a utilização destas bases representa um progresso;
- A utilização de bases permite distinguir entre diferentes conjuntos inconsistentes (um agente pode ser inconsistente);
- Como os conjuntos são grandes entidades (em alguns casos infinitas) para casos práticos as bases deverão ser utilizadas.

8.1.2.1. CONTRAÇÃO DE BASES

A contração parcial é, também, aplicável às bases [52]. A $A \perp p$ chamamos o conjunto de máximos subconjuntos de A que não implicam p . Não é apenas suficiente que não contenham A . Então $\{p \vee q, p \leftrightarrow q\} \perp p = \{\{p \vee q\}, \{p \leftrightarrow q\}\}$. Excluindo a recuperação os postulados são aplicáveis às bases também. Hansson [55] caracterizou a contração parcial nas bases. O operador \div é um operador de contração parcial para um conjunto A se e só se satisfaz os seguintes postulados:

- *Successo*: se $p \in C_n(\emptyset)$ então $p \notin C_n(A \div p)$;
- *Inclusão*: $A \div p \subseteq A$;

- *Relevância*: se $q \in A$ e $q \notin A \div p$ então existe um conjunto A' tal que $A \div p \subseteq A' \subseteq A$ e que $p \notin C_n(A')$ mas $p \in C_n(A' \cup \{q\})$;
- *Uniformidade*: se suporta todos os subconjuntos de A' de A tal que $p \in C_n(A')$ se e só se $q \in C_n(A')$ então $A \div p = A \div q$.

Uma outra abordagem foi proposta com o nome de *contração de núcleo*. Como já visto para qualquer frase p um núcleo- p é um conjunto máximo de implicação de p , ou seja, um conjunto que implica p mas não tem nenhum subconjunto que implique p . O operador de contração pode ser obtido baseando-se no princípio simples de que nenhum núcleo- p seja incluído em $A \div p$. Isto pode ser obtido com uma função de incisão que seleciona pelo menos um elemento de cada núcleo- p para remoção. A operação que consiga remover exatamente aqueles elementos selecionados para remoção pela função de incisão é a operação de contração de núcleo. Todas as contrações parciais em bases são contrações de núcleo, mas existem casos que o contrário não é verdade. Por isso a contração de núcleo é uma generalização das contrações parciais.

8.1.2.2. REVISÃO DE BASES

O operador de expansão para os conjuntos de crenças $-K + p = C_n(K \cup \{p\})$ – foi construído de forma a garantir que seja logicamente fechado. No caso das bases este não é um resultado desejável pelo que a expansão das bases toma um processo distinto. Em cada base A e frase p – $A + 'p$ – a expansão de A por p é o conjunto $A \cup \{p\}$, isto é, $A + 'p = A \cup \{p\}$. Os operadores de revisão para bases podem ser construídos a partir de dois processos: expansão por p e contração por $\neg p$ (a identidade de Levi: $A \div p = (A \div \neg p) + 'p$). De outra forma os processos anteriores podem ser executados por ordem inversa $-A * p = (A + 'p) \div \neg p$. Este processo apenas é aplicável a bases. Estas permitem duas formas distintas para revisão de bases em contração e expansão:

- *Revisão interna*: $A * p = (A \div \neg p) + 'p$;
- *Revisão externa*: $A * p = (A + 'p) \div \neg p$.

A revisão externa por p é a revisão com um estado intermediário inconsistente em que ambos p e $\neg p$ são permitidos, na revisão interna o estado intermediário é um que nem p nem $\neg p$ são permitidos [52].

Revisão Interna

Hansson [56] afirma que o operador $*$ é um operador de revisão interna para uma base A se e só se satisfaz os postulados seguintes:

- *Consistência*: $A * p$ é consistente se p é consistente;
- *Inclusão*: $A * p \subseteq A \cup \{p\}$;
- *Sucesso*: $p \in A * p$;
- *Relevância*: se $q \in A$ e $q \notin A * p$ então existe um qualquer A' tal que $A * p \subseteq A' \subseteq A \cup \{p\}$, A' é consistente, mas $A' \cup \{q\}$ é inconsistente;

- *Uniformidade*: se para todo $A' \subseteq A$, $A' \cup \{p\}$ é inconsistente se e só se $A' \cup \{q\}$ é inconsistente então $A \cap (A * p) = A \cap (A * q)$.

Revisão Externa

Hansson [57] afirma que o operador $*$ é um operador de revisão externa para uma base A se e só se satisfaz os postulados seguintes:

- *Consistência*: $A * p$ é consistente se p é consistente;
- *Inclusão*: $A * p \subseteq A \cup \{p\}$;
- *Sucesso*: $p \in A * p$;
- *Relevância*: se $q \in A$ e $q \notin A * p$ então existe um qualquer A' tal que $A * p \subseteq A' \subseteq A \cup \{p\}$, A' é consistente, mas $A' \cup \{q\}$ é inconsistente;
- *Uniformidade fraca*: se p e q são elementos de A e é verdade para todos $A' \subseteq A$ tal que $A' \cup \{p\}$, é inconsistente se e só se $A' \cup \{q\}$ é inconsistente então $A \cap (A * p) = A \cap (A * q)$;
- *Pré-expansão*: $A + p * p = A * p$.

8.2. ARGUMENTAÇÃO

Argumentação pode ser vista como a disciplina que estuda como chegar a conclusões através de um raciocínio lógico. Em si podemos dizer que a argumentação pode ser o processo que nos leva uma decisão. Este processo poderá ser feito pelo próprio para tomar uma decisão mais acertada relativamente a um determinado problema ou sobre um terceiro para o fazer ver da validade, ou não, da sua decisão. Desde há muito, do tempo dos filósofos e retóricos gregos, que a argumentação é estudada na pesquisa dos requerimentos que fazem um argumento correto. Por diferentes disciplinas como a filosofia, psicologia, neurociência, linguista, para citar algumas. Aristóteles ainda influencia a IA, pensemos nos silogismos [58] que são um pequeno exemplo. De [59] verifica-se a existência de quatro tarefas principais associadas:

- *Identificação*: reconhecer as premissas e conclusão de um argumento e então concluir se pertencem ao mesmo esquema de argumentação;
- *Análise*: identificar as possíveis premissas implícitas ou conclusões que precisam de ser explicitadas para que o argumento possa ser avaliado;
- *Avaliação*: pretende verificar se o argumento é fraco ou forte;
- *Construção*: criação de novos argumentos que suportem a conclusão específica.

Um argumento pode ser definido como um conjunto de proposições, compostas de três partes: conclusão, conjunto de premissas e inferência das premissas para a conclusão. A noção de *Revogabilidade*, introduzida por [60] e [61], é pertinente com os argumentos. Quando a conclusão, suportada por regras, é revogada por nova informação, este raciocínio é revogável. E quando uma conclusão é alcançada por esta forma temos argumentos e não provas. Um argumento pode sustentar outros argumentos, mas também atacar outros argumentos. Isto leva à questão de ataque/refutação. A partir de [59] um argumento pode ser atacado de várias formas. Pode ser atacado por outro se uma questão levanta dúvidas sobre a sua aceitabilidade, no qual este fica em suspenso até à sua resposta. Uma outra forma é questionar as premissas em que se baseia. Ou apresentar um contra-argumento que suporta a conclusão oposta ao argumento original.

Em [62] os *Sistemas Argumentativos* apresentam uma forma de formalizar o senso comum. Existem vários trabalhos introdutórios para modelos lógicos para suporte da argumentação [63] [64] [65]. O mais influente é a plataforma abstrata de Dung [66]. Apresenta uma noção importante na aceitabilidade dos argumentos e a argumentação é vista como uma forma especial de programação lógica.

8.2.1. PLATAFORMA ABSTRATA DE DUNG

Na proposta de Dung [66] a noção de argumento é apresentada de forma abstrata para permitir uma possível expansão da mesma. Nesta bordagem a plataforma de argumentação é um par: $\mathcal{AF} = \langle \mathit{Args}, \mathit{attack} \rangle$. Args é o conjunto de todos os argumentos possíveis e attack é a relação binária em Args . Se $(A, B) \in \mathit{attack}$ significa que o argumento A ataca o argumento B . A relação entre os argumentos é que se o argumento A ataca o B o primeiro será refutado se é possível descobrir pelo menos um atacante que não foi refutado. A aproximação assume a existência de um conjunto de argumentos ordenados pela relação binária de refutação. É possível definir diversas noções de extensões para os argumentos que permitam capturar diversos tipos de consequências revogáveis.

8.2.2. DELP

O trabalho apresentado em [67] DeLP (*Defeasible Logic Programming*) lógica programável revogável. O formalismo associa a Lógica Programável e a Argumentação Revogável e apresenta uma evolução a partir de [64]. A informação pode ser representada em forma de regras fracas e alcança a conclusões garantidas através da utilização de um argumento de inferência de argumentação revogável. A plataforma apresenta três elementos: um conjunto de factos, um conjunto de regras estritas e um conjunto de regras revogáveis. Os factos apresentam um literal, ou seja, um elemento básico ou sua negação. Uma regra estrita apresenta um par ordenado $\mathit{Head} \leftarrow \mathit{Body}$ em que o primeiro é um literal e o Body é um conjunto finito de conjunto não vazio de literais. Uma regra revogável apresenta um par ordenado $\mathit{Head} < \mathit{Body}$ em o primeiro é um literal e o Body é um conjunto finito de conjunto não vazio de literais. Um programa de DeLP é um conjunto de factos, regras estritas e revogáveis. A *Derivação Revogável* apresenta a derivação de um literal \mathcal{L} de \mathcal{P} anotada como revogável se existe informação em contradição com \mathcal{L} que previne a aceitação de \mathcal{L} como conclusão válida. A *Derivação Estrita* promove-se através de h se h é um facto ou todas as regras utilizadas para obter a Derivação Revogável de h são regras estritas. Regras contraditórias ocorrem se e só se existe uma derivação revogável para um par de literais contraditórios do conjunto.

8.3. REVISÃO DE CRENÇAS E ARGUMENTAÇÃO

Em [68] foi apresentada uma exploração da relação entre estes dois tópicos através da visão de que ambos são complementares. E não é possível construir um modelo de decisão sem o suporte de ambas.

8.3.1. SISTEMAS DE MANUTENÇÃO DE VERDADE

Em 1979 Doyle apresentou o *Truth Maintenance System* (TMS) que é um método para representação de crenças e justificações [63]. Este pretende manter a consistência do sistema quando chega nova informação. O TMS associa uma estrutura de dados especial (nós) com os dados de resolução de problemas (entradas de bases de dados, regras de inferência, procedimentos) e guarda as justificações (argumentos) para potenciais crenças inferidas e disponibiliza o corrente conjunto de crenças. O processo inicia-se quando a informação nova é adicionada ao nó; depois é processada através dos passos do processo de argumentação. Outro tipo proposto, Kleer [69], é um sistema baseado em assunções do TMS. Este sistema TMS não avalia as justificações, mas relaciona os nós de decisão com o correspondente conjunto de assunções que representa o contexto que as suporta. Este conjunto é derivado pelo ATMS (*Assumption-Based Truth Maintenance System*) das justificações fornecidas pelo nó de resolução. Desta forma as crenças podem ser derivadas e utilizadas como argumentos. As assunções são consideradas como dados primitivos e os outros dados podem ser de si derivados. Sendo a consistência não essencial no geral.

8.3.2. REVISÃO DE CRENÇAS E EPISTEMOLOGIA

Pollock e Gillies [70] estudaram a dinâmica dum sistema de revisão de crenças considerando as relações entre as crenças e tentaram derivar uma revisão de crenças de uma mais concreta teoria epistemológica. Afirmam que um dos objetivos da revisão de crenças é a geração de uma base de conhecimento em que cada peça de informação tem uma justificação (por percepção) ou garantida pelos argumentos que sustentavam as crenças anteriores. A consequência é que os conjuntos de crenças justificadas podem exibir várias incoerências lógicas devido ao estado intermédio de raciocínio que apresentam. Neste caso a teoria proposta é sustentada pela garantia ao invés das justificações. Estas garantias têm em conta todas as possíveis inferências para que apenas exista uma forma de adquirir novas garantias, o que é conseguido através da percepção.

8.3.3. REVISÃO DE CRENÇAS E EXPLANAÇÕES DEDUTIVAS

Fallapa *et al.* [71] apresentou um operador de revisão sem-prioridade baseado no uso de explicações. Sendo a ideia principal que um agente tem de requerer explicações que suportem a nova informação inconsistente antes de a incorporar. Esta inconsistência deve ao seu conhecimento. Os autores elaboram na distinção clássica entre *explanandum* (conclusão final) e *explanans* (frases que suportam a conclusão). A estrutura de uma explicação é similar à estrutura de um argumento dedutivo em que a distinção reside no facto de cada crença de uma explicação é sólida/defendível onde algumas crenças de um argumento podem ser derrotadas. Cada explicação contém, regras e conhecimento, se as frases na *explanans* são mais plausíveis que as frases na base de crenças então a explicação é incorporada. Para o processo de mudança as crenças que suportam as explicações são o elemento utilizado. A revisão parcial e de núcleo são considerados por um conjunto de frases e são apresentados

com teoremas de representação. Os operadores podem, parcialmente, aceitar a nova informação pelo que não são priorizados.

8.3.4. REVISÃO DE CRENÇAS ORIENTADA A DADOS

Pagliari e Castalfranchi [72] uniram, na mesma estrutura, as duas visões a partir do desenho de Taoulmin. A ligação à revisão de crenças é feita considerada uma persuasão levada a cabo pela argumentação. Então esta é considerada como iniciadora do processo que deverá ser sucedido. A visão orientada a dados foi proposta como alternativa ao AGM. Dados e crenças são as categorias básicas do elemento de informação, para efetuar a distinção entre elementos de que são apenas adquiridos e armazenados pelos agentes e elementos de informação que os agentes consideram como representação fiel do mundo (estado). A distinção entre dados e crenças. As crenças são um subconjunto de dados significando que um agente pode ser consciente de um estado que não admite como crença pois poderá não ser fiável o suficiente. As estruturas de dados são redes de nós (dados) interligadas pelas relações de suporte, contraste e união. Os dados podem ser selecionados/rejeitados baseados nas suas propriedades tais como relevância, credibilidade, importância e similitude. A união dos dados e garantia suporta a reivindicação e a garantia é suportada por estes e contrastada pela refutação. Logo esta torna a garantia menos fiável.

8.3.5. REVISÃO PRIORITÁRIA POR ARGUMENTOS

Em Rostein *et al.* [73] é proposta uma teoria abstrata em que as dinâmicas da plataforma de argumentação são captadas através dos princípios de Revisão de Crenças. Esta é definida pela inclusão constrangimentos de dialética. Os processos de expansão, contração e revisão são incorporados nesta plataforma. Sendo a revisão expressa em termos de expansão e contração. A teoria permite a introdução de argumentos assegurando que será sustentado mais tardiamente. O operador de expansão é utilizado diretamente, mas o de contração abre um leque de possibilidades: pode afetar um determinado número de argumentos no sistema para que este se mantenha sem perturbações de maior (princípio de mudança mínima AGM). Moguillansky et al. instanciou-se estas operações para DeLP. O operador WPA – *Warrant-prioritized argument revision* – que implementa a mudança foi definido. Quando existe uma revisão por um argumento $\langle A, \alpha \rangle$ após a revisão será tal que A é um argumento defendível e α será uma garantia. A questão é que o operador é falacioso na seleção de argumentos bem como as incisões feitas sobre estes. Um critério utilizado na seleção de argumentos determina quais os que não devem ser considerados e, quando a seleção é feita, as incisões farão que esses desapareçam de acordo com o princípio mudança mínima escolhido.

8.3.6. RECUPERAÇÃO E REINSTALAÇÃO RELACIONADA

Boela *et al.* [74] tentaram a ligação direta entre Argumentação e Revisão de Crenças no nível de propriedades abstratas. Consideraram a Argumentação como persuasão a creditar e a persuasão deve

ser relacionado com a Revisão de Crenças. A relação entre recuperação e reinstalação foi estabelecida através de referência da reinstalação como uma situação em que um argumento que não foi aceite por causa da existência de argumento que se sobrepôs torna-se novamente aceitável devido à confrontação de um novo argumento com o sobrepujante. De acordo com a recuperação a expansão por α deve recuperar o que foi perdido quando α foi contraído (princípio de mudança mínima AGM).

8.4. CONCETUALIZAÇÃO DO MODELO

Em [68] Falappa *et al.* é apresentada uma visão concetual do modelo em Argumentação e Revisão de Crenças e com possíveis ligações entre os dois campos. A partir dos passos básicos de raciocínio e assumindo o estado epistemológico representado numa determinada plataforma. A informação recebida \mathcal{J} poderá ter diversas formas. Pode ser um simples facto proposicional (âmbito do AGM) ou ser mais complexo com um grau de plausibilidade ou uma regra ou um argumento completo ou, até, um conjunto de tais entidades.

Para processar \mathcal{J} é necessário reconhecer a sua origem. Pois este conhecimento irá afetar a sua disponibilidade para a receber e adotar. Se for baseado na observação do agente este está convencido da veracidade desta informação. Mas esta poderá advir da comunicação de outro agente ou de outro meio de comunicação. Sedo assim o agente terá de requerer a justificação pelo que, baseado nas suas crenças, irá avaliar \mathcal{J} e a justificação para que decida da validade de aceitação de \mathcal{J} . Caso seja adotado \mathcal{J} o agente irá aplicar os processos para a sua incorporação, de forma consistente, nas suas crenças. Promovendo as técnicas de Revisão de Crenças para alterar o seu estado epistemológico. Por fim, a partir do seu novo estado, irá derivar as suas crenças plausíveis. Estes passos podem ser aplicados a questões/perguntas que o próprio agente deva responder:

- Receção de informação;
- Avaliação da informação;
- Mudança de Crenças;
- Inferência.

Como visto do processo anterior a Argumentação pode contribuir para o passo de avaliação e a Revisão de Crenças para o passo de mudança de crenças. Pode parecer simples, mas o passo de avaliação poderá incluir mudanças hipotéticas, considerando o que aconteceria se a nova informação fosse aceite, e a revisão de crenças baseia-se nas ligações lógicas estabelecidas entre informação que pode ser representada por argumentos. As crenças plausíveis podem ser obtidas tanto da Argumentação como através da Revisão de Crenças. Sendo que estas preocupam-se com a dinâmica do processo de raciocínio. Os operadores de revisão podem ser aplicados sobre diversos aspetos (crenças, intenções, preferências, teorias, entre outros pesquisa) e a Argumentação pode ser utilizada para negociar, pesquisa, inquirição o que reflete uma complexa, e interrelacionada, visão sobre ambas as disciplinas.

8.4.1. COMPARAÇÃO

De [68] as diferenças entre as duas disciplinas parecem prevalecer como exemplo os fundamentos de sintática e semântica de ambas. A representação do conhecimento utiliza as formas lógicas da Revisão de Crenças. Em casos de plataformas mais avançadas e complexas a semântica da lógica clássica é utilizada. A Argumentação debruça-se na interação de argumentos como peças informação que podem abater e/ou rebater outros argumentos. Estes não são homogêneos apresentado uma diversidade de formas, desde estruturas complexas de argumentos a objetos abstratos (sem estrutura interna – Dung). Esta visão pretende resolver os conflitos existentes, baseando em fundamentos lógicos (contradições) e utilizando relações de preferência. A Revisão de Crenças proporciona uma plataforma declarativa, baseada em postulados, enquanto a Argumentação preocupa-se mais com as técnicas de justificação.

A partir dos postulados do AGM obtém-se uma fundamental visão explícita na plataforma utilizada. A teoria que se baseia em conjunto fórmulas dedutivamente fechadas (representa os conjuntos de crenças) parece não ter relação com a teoria da Argumentação. Assim os trabalhos da revisão de bases ou iteração dos estados epistemológicas fornecem uma base melhor para a comparação pois permitem ter uma visão mais aprofundada dos processos de mudança. Na Argumentação as aproximações baseadas em regras, ou derivação das mesmas, fornece ligações lógicas entre o que é pressuposto e o concluído. Neste ponto a Revisão de Crenças mantêm-se num nível abstrato, descrevendo por axiomas o que são inferências boas/corretas/plausíveis. Por sua vez a Argumentação preocupa-se com o como e porquê das conclusões alcançadas. Os autores alegam que estas observações fazem das duas áreas complementares.

8.4.2. ARGUMENTAÇÃO NA REVISÃO DE CRENÇAS

Alguma das técnicas da Argumentação podem ser utilizadas na Revisão de Crenças [68]. A abordagem inicial do Doyle (TMS baseada em justificações) [63] estudou as iterações entre as justificações e a nova justificação adicionada para revelar quais as conclusões justificadas. Kleer [69] no ATMS é mais centralizado em manter as assunções ao invés de implementar os processos de mudança. No DBR [72] Paglieri e Castlefranchi existe os dados e as crenças. Sendo estas aceites em determinados contextos representados por conjunto de assunções e que não é necessária ser consistente em toda a base de dados. É simples referir-se a vários contextos e mover-se para diferentes pontos no espaço de pesquisa representado pelo contexto próprio relevante. O TMS e ATMS em bases de crenças são aproximações iniciais à Revisão de Crenças.

Em [71] combinaram a revisão de bases com a ideia do ATMS e propuseram um sistema que utiliza as estruturas argumentativas em forma de explicações para revisão não prioritária de bases de crenças. A nova informação consiste numa proposição e as razões para a aceitar (regras e pré-requisitos dos quais a proposição pode ser derivada dedutivamente). Para a integração da avaliação argumentativa

da nova afirmação no processo de revisão os autores definiram um operador de aceitação parcial de revisão com os seguintes passos:

- A entrada epistemológica é o conjunto de frases A como *explanans* do *explanandum* α ;
- A é junta a K (produzindo, talvez, um estado intermédio inconsistente)
- Todas as possíveis inconsistências de $K \cup A$ são removidas retornando um estado consistente da revisão de base $K * A$. O operador é um operador de revisão externa (esta toma parte fora do conjunto original).

A aceitação de α não depende da avaliação do seu *explanans* A de acordo com a corrente base de crenças. A aceitação do *explanans* força a aceitação do *explanandum* no conjunto revisto (explicação é baseada na dedução clássica). Contudo, enquanto a *explanans* possa ser explicitamente incluído no conjunto revisto, o *explanandum* pode ser inferido desse sem ser incluído. Logo a distinção entre informação explicitamente dada e crenças inferidas é respeitada e pode ser implementada apenas quando trabalhando com bases de crenças ao invés de conjunto de crenças.

Os autores reivindicam que existe uma diferença entre o processo de revisão por um conjunto de crenças e o processo de argumentação. Na revisão as crenças externas são comparadas com as internas e, pós um processo de seleção, algumas são descartadas enquanto outras são aceites. Além de na Argumentação o processo ser mais procedimental. Um argumento é rebatido pelos contra-argumentos, defende-se por contra-argumentos com os primeiros e assim por diante. O racional do operador é que se identifica com o argumento principal na razão de acreditar nas razões pela qual a nova informação é avaliada e não na informação em si.

8.4.3. REVISÃO DE CRENÇAS NA ARGUMENTAÇÃO

Os métodos da Revisão de Crenças podem implementar características dinâmicas na plataforma de argumentação como afirmado em [68]. Rotstein *et al.* [73] e Moguillansky *et al.* [75] apresentam aproximações para sustentar a revisão em Sistemas Argumentativos. Existem diversas formas de aplicar a Revisão de Crenças na Argumentação [68]:

- Alteração por adição ou subtração de um argumento ou conjunto de argumentos;
- Alteração da relação de rebate entre argumentos (aceitação/refutação);
- Alteração do estado das crenças (como conclusão dos argumentos);
- Alteração do tipo de argumento (estrito para anulável e vice-versa).

A distinção ente adição/subtração de um argumento, ou de um conjunto de argumentos, é similar à distinção entre uma mudança (modelo AGM) e múltiplas mudanças (como múltiplas contrações [76]). Estas podem despoletar mudanças nas conclusões justificadas e podem levar a uma revisão de base que lida com a mudança de bases argumentativas, nas quais cada dedução é substituída por um processo de argumentação. O método de contração de núcleo utilizado para eliminação de elementos da base (funções de incisão) tem grande interesse [75].

A alteração ente a relação aceitação/refutação entre argumentos pode levar a um diferente funcionamento do sistema. A compreensão e controlo do que constitui desafios substanciais para a

teoria de Revisão de Crenças. *Boella et al.* [77] propôs uma aproximação que ataca a relação numa forma ditatorial em favor da nova informação. É possível definir processos de mudança mais complexos utilizando as ideias da mudança epistémica que trata da modificação de relação entre possíveis mundos. Os argumentos são distintos de mundos possíveis, mas estes podem ser considerados como aproximações para realizar processos de mudança mínima em relações gerais. A mudança de estado de crenças pode ser consequência das mudanças no sistema de argumentações. Como visto numa plataforma de argumentação onde os argumentos são baseados em conclusões, [67], a adição de um argumento pode alterar o estado de uma reivindicação. As preocupações com o aspecto de raciocínio da Revisão de Crenças podem ser consideradas e investigações ao nível dos postulados podem ser úteis. Um exemplo será a relação entre reinstalação na Argumentação e recuperação em Revisão de Crenças proposto em [74].

A mudança de estado de um argumento de estrito para refutável, ou vice-versa, equaciona um momento novo na Revisão de Crenças. A ideia é que inconsistências podem surgir quando nova informação tem de ser incorporada no conjunto de crenças pode ser eliminada não só pela remoção de argumentos (crenças respetivas) mas também por mudar crenças estritas em regras refutáveis (condicionais respetivas). A possibilidade de alteração do estado das crenças introduz nos operadores de revisão uma nova característica, com importância para a argumentação, pois os argumentos são, muitas vezes, formados por crenças refutáveis.

Em [71] foi introduzido um novo tipo de revisão de base que implementa uma classificação dinâmica de crenças que é relevante para a Argumentação e Revisão de Crenças. A plataforma proposta inclui condicionais refutáveis gerados pela estrutura de revisão de crenças compostas regras refutáveis e conhecimento irrefutável e providencia um mecanismo para qualificar dinamicamente as crenças como refutáveis ou irrefutáveis. Promovendo um conjunto de atitudes epistémicas mais completo e estendendo o poder de inferência dos sistemas baseados em conhecimento. Em [78] Falappa *et al.* propôs a extensão da aplicação deste operador de revisão não prioritário ao DeLP [67].

8.5. MODELO BDI

O modelo BDI (*Delief-Desire-Intention*) providencia uma arquitetura para o desenvolvimento de agentes inteligentes. Proposto por Bratman [79] como uma teoria para raciocínio prático para explicar o comportamento utilizando crenças, desejos e intenções como atitudes mentais. Como descrito em [80] a assunção básica do modelo é que as ações derivam de um processo chamado raciocínio prático. Este consiste em dois passos [81]:

- Definir objetivos ao selecionar, de um conjunto de desejos, que têm de ser atingidas de acordo com as crenças correntes do agente – Deliberação;
- Determinação de como atingir os desejos através dos recursos disponíveis – Raciocínio de recursos-objetivos.

As três atitudes mentais descrevem-se como [80]:

- *Beliefs* (Crenças): representam as características do ambiente que são atualizadas de forma apropriada após cada interação ou por percepção. Podem ser descritas como as componentes informativas do sistema (o conhecimento do agente);
- *Desires* (Desejos): representam os objetivos a atingir. Podem incluir prioridades e custos associados a cada um destes. Podem representar o estado de motivação do sistema;
- *Intentions* (Intensões): representam o plano de ação corrente. A parte deliberativa do sistema. Esta poderá incluir outros planos.

Em 1995, Rao e Georgeff [82] adotaram o modelo BDI para o software de agentes ao apresentar a teoria formal e um interpretador abstrato da BDI que é a base de quase todos os sistemas BDI. Este opera sobre as crenças, planos e objetivos do agente sendo a maior distinção os objetivos um conjunto consistente de desejos que podem ser atingidos em grupo evitando a fase mais complexa de deliberação. O interpretador realiza, através da seleção e implementação de planos para um específico propósito ou evento, o processo de raciocínio de recursos-objetivos. Sendo esta a sua principal tarefa.

Wooldridge apresentou uma estrutura genérica [83] da arquitetura BDI para representar o aspeto prático de raciocínio de um agente. Da Figura 8.1 destaca-se os sete componentes de um agente BDI:

- Um conjunto de crenças, representando a informação que o agente tem do seu ambiente;
- Uma função de revisão de crenças, que determina um novo conjunto de crenças baseado na percepção das entradas e das crenças do agente;
- Uma função de geração de opções, que determina as opções disponíveis ao agente (desejos) baseados nas suas crenças sobre o ambiente e nas suas intensões;
- Um conjunto de opções (desejos), que representam o processo de deliberação do agente baseado nas suas crenças, desejos e intensões;
- Um conjunto de intensões, que são o corrente foco do agente, ou seja, os estados que o agente está determinado a atingir;
- Uma função de seleção de ações, que determina a ação a tomar baseado nas intensões correntes.

É possível a implementação de agentes deliberativos complexos através de um agente básico com uma estrutura em ciclo associada à plataforma BDI. Menciona-se os pontos seguintes entre as limitações e críticas ao modelo:

- **Aprendizagem:** os agentes BDI não apresentam algum mecanismo que possibilite esta;
- **Múltiplos agentes:** o modelo BDI não especifica mecanismos de interação entre agentes e a integração num sistema multiagentes;
- **Objetivos explícitos:** a maioria das implementações não apresentam uma representação explícita dos objetivos.

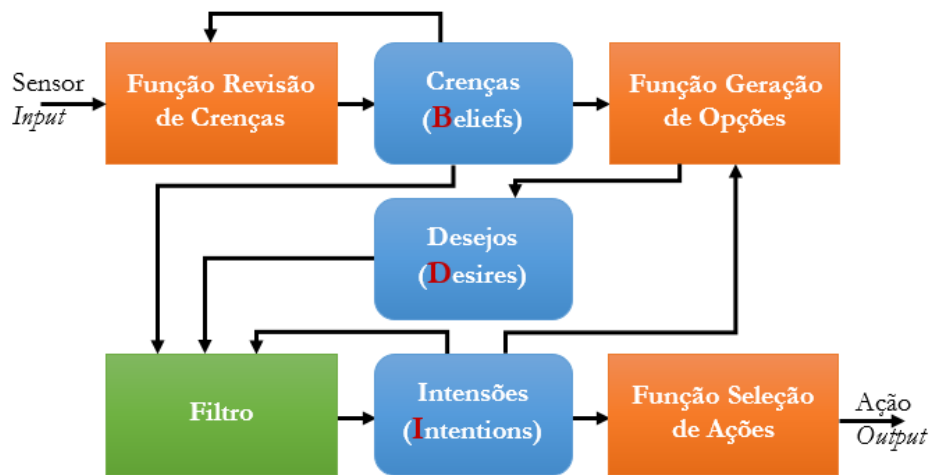


Figura 8.1 – Estrutura BDI genérica

Em [78] apresenta-se a necessidade do modelo BDI básico ser complementado com dois mecanismos: um para o raciocínio sobre intensões e outro para a revisão de crenças sobre as suas percepções.

8.6. CONCLUSÃO

Neste capítulo abordamos duas ferramentas interessantes. Após termos discutido a representação do conhecimento e raciocínio, que poderiam ser complementados com ferramentas que permitissem a revisão da informação presente estas permitem esse objetivo. Ou seja, a partir de uma *KB* com extensão de estas ferramentas, Revisão de Crenças e Argumentação, poderemos realizar a modificação do conhecimento representado.

9. TEORIA DA DECISÃO

Uma discussão adaptada a partir de [84] que apresenta como são tomadas decisões em conjunto. Um ramo da estatística, *Teoria da Decisão*, que é produto dos esforços de economistas, matemáticos, filósofos e cientistas sociais, no sentido de fazer sentido de como indivíduos e grupos tomam ou devem tomar decisões. As decisões tanto em grupo como individuais envolvem uma escolha entre duas ou mais opções (opiniões, ações, entre outros) em que cada uma irá produzir um determinado resultado. Além de que cada opção depende do ambiente sendo que o resultado pode não ser o esperado quando a decisão foi tomada. Assim há que ter em conta três componentes: ações, estados e resultados. O resultado dependerá das ações tomadas num determinado estado. Este estado é um tanto ou quanto abrangente pois pode incluir várias características entre físicas e não físicas. Num problema de decisão o decisor tem de determinar o conjunto relevante de ações, estados e resultados para a caracterização do problema. Este é a especificação do problema. Esta especificação é importante pois cada problema pode gerar várias especificações. É da responsabilidade do decisor a aplicação da correta especificação do problema sob pena de não tomar uma decisão correta ou de ter resultados inesperados.

Só uma cuidadosa análise pode permitir que as especificações sejam mutuamente exclusivas. É possível garantir, de forma simples, esta exclusividade introduzindo aos possíveis estados a opção de que nenhum deles se aplica. Como a opção “nenhum dos anteriores” em diversos questionários. Esta pode levar a escolhas incorretas pois não foram consideradas todas as possibilidades. Na escolha das especificações do problema existem três implicações interessantes. A primeira tem a ver com a própria descrição dos estados. Qualquer problema integra alguns resultados que o decisor considera melhor que outros. Pelo que a descrição dos mesmos pode simplificar-se a resultado *bom* e resultado *mau*. Este denomina-se por princípio do domínio. A ação *A* domina a ação *B* se, na comparação estado a estado, o *A* fornece resultados que são pelo menos tão bons quando aqueles fornecidos por *B*. Esta regra diz que se um estado domina os outros escolhamos este. Agora, nem sempre nos podemos guiar por este princípio pois pode nos levar a considerar que a escolha já foi feita pois o estado que domina os outros é por ser a escolha correta. Neste caso, aparentemente não existe a necessidade de escolher pois à partida a escolha parece estar feita. Aprofundando o assunto vê-se que o princípio só se aplica quando as ações não afetam as probabilidades dos estados.

Outro caso interessante na especificação é a escolha da própria pois ao escolher as especificações podemos aplicar a teoria da decisão. Ora neste caso teremos de fazer uma escolha que por sua vez pode ser sujeita à teoria da decisão. Este é um caso de regressão pois poderemos aplicar a teoria da decisão a todas as escolhas que forem feitas. Um outro caso pode ser exemplificado considerando dois indivíduos, *A* e *B*, que competem num determinado negócio: O *A* faz um estudo aprofundado sobre se vale a pena fazer o investimento chegando à conclusão que não. Enquanto o *B*, sem qualquer

preparação, decide investir. Isto quer dizer que o indivíduo *A* fez uso da Teoria da Decisão enquanto o *B* bem poderia ter utilizado uma moeda para realizar a decisão. A surpresa vem quando o indivíduo *B*, sem ter feito qualquer preparação, decidiu investir e o resultado é que este acaba por fazer um grande negócio. Assim é possível dizer que a Teoria da Decisão foi irracional? Para resolver esta situação podemos distinguir entre a escolha correta e escolha racional. Na tomada de decisões são baseadas naquilo que pensamos que pode acontecer ou naquilo que é mais provável acontecer e não podemos ter a certeza que o resultado será o melhor possível. Um agente fará uma decisão correta se o resultado obtido for pelo menos tão bom como todos os outros possíveis resultados depois da sua ação. Teremos sempre de tomar uma decisão baseada na informação disponível e da avaliação do risco envolvido tomando uma decisão racional.

Por vezes é possível ter a certeza do resultado das decisões. Mas geralmente apenas é possível estimar uma probabilidade dos resultados possíveis. O primeiro caso é designado por escolha sob certeza. No caso de ter as probabilidades de cada resultado para cada ação é denominada decisão sob risco. Quando não é possível atribuir probabilidades, ou não faz sentido, aos possíveis resultados e denominada decisão sob ignorância. Esta pode ser total ou parcial. Sendo esta distribuição idealizada pois os casos podem apresentar várias características. A formulação dos problemas pode ter um efeito na decisão pois considera-se o par ação-estado. Mas o resultado pode depender da escolha bem como do estado. Este pode ter em si uma certa incerteza. O que provoca a necessidade de uma melhor formulação do problema substituindo a formulação das decisões/estados ou particularizando os estados para ter sempre o par ação-estado. Esta formulação leva às probabilidades incondicionais e condicionais. Sendo a primeira a probabilidade de um resultado independente da ação tomada uma doença que qualquer pessoa pode ter mesmo são comportamento de risco. As probabilidades condicionais são utilizadas num caso que a probabilidade de um determinado resultado varia conforme a ação tomada. Sair coroa no lançamento de uma moeda é independente de se ter escolhido coroa. Mas ter boas notas não é independente da decisão de estudar mais. Pelo que é necessário a correta formulação do problema para as corretas decisões.

A Teoria de Decisão que se foca nas decisões envolvendo um agente e suas escolhas baseadas nas suas preferências e ambiente é individual. Existem situações em que vários agentes são ativos da determinação dos resultados. Designa-se por *Jogo*. Cada jogo tem dois ou mais jogadores, existem movimentos em que um ou mais jogadores têm de realizar uma escolha cada um. Se existe um elemento de sorte no jogo esta conta como mais um jogador que toma decisões aleatórias. As regras de cada jogo determinam para cada sequência de movimentos se tem um resultado e qual é. Uma sequência de movimentos que determinam um resultado é chamada de jogada.

O conceito de *Teoria de Jogos* abstrai a maior parte dos elementos psicológicos e das características dos jogos. Um jogo de Xadrez com peças de marfim ou de plástico continua a ser um jogo de Xadrez [84].

9.1. ESCOLHA SOCIAL

A teoria *Escolha Social* é um estudo de processo de decisões e procedimentos. Não apenas uma única teoria, mas um conjunto de modelos e resultados da agregação de várias entradas. Tais como decisões individuais e coletivas. As questões centrais são: como pode um grupo de indivíduos decidir numa escolha entre várias opções? Quando é um sistema de votação democrático? Como pode um conjunto de elementos decidir de forma coerente sobre assuntos mesmo tendo em conta as preferências individuais dos elementos? Como é possível escalonar entre várias alternativas?

De uma forma mais abstrata o problema da escolha social pode ser visto um grupo, de indivíduos tem duas ou mais alternativas coletivas ou políticas para adotar. Os membros do grupo têm as suas próprias preferências relativamente à escolha coletiva. O problema surge em tentar desenvolver uma escolha coletiva desta situação. Iniciada no séc. XVII por Nicolas de Condorcet e Jean-Charles de Borda e desenvolvida por Charles Dogson (conhecido na literatura por Lewis Carrol) teve um maior desenvolvimento no séc. XX através do trabalho de Kenneth Arrow, Amartya Sem e Duncan Black. A sua influência estende-se através da economia, ciência política, filosofia matemática e, mais recentemente, biologia e ciência da computação. Além contribuir para o conhecimento dos procedimentos de escolha coletiva tem aplicações no desenho institucional, economia e epistemologia social. A escolha social pode ser vista em dois passos: combinar a preferência dos elementos do grupo para obter uma preferência coletiva e combinar as probabilidades das suas preferências para obter uma função de probabilidade coletiva.

As principais figuras associadas com o desenvolvimento da teoria foram Nicolas de Condorcet (1743-1794) e Kenneth Arrow (nascido em 1921). O primeiro foi um pensador liberal na altura da Revolução Francesa que foi perseguido pelos revolucionários pelas suas críticas. No seu ensaio *Essay on the Application of Analysis to the Probability of Majority Decisions* (1785) advoga um sistema particular de votação, votação por maioria de pares, e apresenta duas das suas mais proeminentes observações. A primeira, *Condorcet's jury theorem*, é que se cada membro de um júri tem uma hipótese igual e independente melhor que aleatória, mas pior que perfeita, de fazer um julgamento se o arguido é ou não culpado (ou outra qualquer proposição factual) a maioria dos jurados é capaz de dar um veredito correto que cada jurado individualmente. A probabilidade de um julgamento correto aproxima-se de 1 com o aumento do número de jurados

A segunda observação, *Condorcet's paradox*, é que a decisão da maioria pode ser irracional (intransitiva) mesmo quando as preferências individuais são racionais (transitivas). Um terço de um grupo prefere X e Y a Z , o outro terço prefere Y a Z a X e o final prefere Z a X a Y . Existem duas maiorias (dois terços) que preferem X a Y , Y a Z e Z a X o que cria um 'ciclo' violando a transitividade. Condorcet antecipou um problema chave da Escolha Social moderna; a regra da governação por maioria é, ao mesmo tempo, um método plausível de decisão coletiva e sujeita a alguns problemas interessantes. Sendo a resolução destes problemas uma parte do trabalho da Escolha Social.

Condorcet investigou um método particular de votação (maioria) Arrow, que venceu o Nobel em Economia em 1972, introduziu uma aproximação geral ao estudo da agregação de preferências. Influenciado pelo seu professor de lógica, Alfred Tarski (1901-1983). Arrow considerou uma classe de métodos de agregações possíveis, que designou de funções de benefícios sociais, e inquiriu quais delas satisfaziam certos critérios ou desejos. Demonstrou que não existem métodos para agregar as preferências de dois ou mais indivíduos acerca de três ou mais alternativa em preferências coletivas. Este resultado, conhecido como *Arrow's impossibility theorem*, promoveu muito debate tanto na Escolha Social como na economia de benefícios sociais.

Hoje em dia a maioria moveu-se além das interpretações negativas deste resultado e estão mais interessados nas aproximações envolvidas na procura de processos de decisão satisfatória. Esta visão, ainda mais que a original, promoveu a influência de Arrow. O resultado paradoxal moderno é o teorema da caracterização em que o objetivo é identificar um conjunto de condições suficientes e necessárias que univocamente caracterizem uma determinada solução (ou classes de soluções) de um tipo particular de decisão coletiva de um problema.

Jean-Charles de Borda, contemporâneo de Condorcet, defendeu um sistema de votação visto como uma possível alternativa. A *Contagem de Borda*, definida mais tarde, evita o paradoxo de Condorcet mas viola uma das condições de Arrow. Este debate entre diferentes visões não é recente pois, já na Idade Média, Ramon Llull (1235-1315) propôs um método de agregação para a votação por pares enquanto Nicolas Cusanus (1491-1464) propôs uma variação da Contagem de Borda (McLean 1990). Em 1672 Samuel von Pufendorf (1632-1649) fez comparação de maioria simples, maioria qualificada e regras de unanimidade e desenvolveu uma análise da estrutura de preferências que pode ser vista como precursora (Gaertner 2005). No séc. XIX Charles Dodgson (1832-1898) redescobriu as observações de Condorcet e Borda e desenvolveu a teoria de representação proporcional. Estas observações foram trazidas a lume através do trabalho de Duncan Black (1908-1991) para a comunidade científica. Este também fez várias observações relacionadas com regra da votação maioritária. George Thédude Guilbaud escreveu um artigo, 1966, revisitando a teoria de Condorcet através de uma perspectiva lógica [85].

9.2. ESCOLHA SOCIAL COMPUTACIONAL

A teoria de Escolha Social preocupa-se como o desenho e análise métodos para o processo de tomada de decisões coletivas. O campo da Inteligência Artificial tem visto esta teoria com interesse ao longo dos tempos.

Existem dois campos: um importa as noções e métodos da inteligência artificial para resolver questões que emergem da escolha social enquanto o outro faz o percurso inverso; importa os conceitos e procedimentos da escolha social para a resolução de problemas da inteligência artificial. A partir de [86] tiramos um conjunto de tópicos abordados pela Escolha Social Computacional.

Agregação de preferências

A agregação de preferências significa mapear uma coleção $P = \langle P_1, \dots, P_n \rangle$ de relação entre preferências (ou perfis) de agentes individuais numa relação de preferência P^* (que implica circular o Teorema da Impossibilidade de Arrow em relaxar uma das suas condições). Por vezes apenas se pretende a determinação de uma alternativa socialmente preferida, ou um subconjunto de alternativas sociais preferidas em vez de uma relação de preferência coletiva: uma *função de escolha social* mapeia um perfil coletivo P numa única alternativa enquanto uma correspondência de escolha social mapeia um perfil coletivo P num subconjunto, não vazio, de alternativas.

Teoria de Votação

O ato de votar é um dos meios mais populares de atingir decisões comuns. Esta área levou ao estudo de uma variedade propriedades de várias famílias de regras de votação, mas, tipicamente, negligenciou a área computacional. Um panorama de regras e votação foi proposto na área: A regra de votação posicional computa um valor para cada candidato a partir de cada perfil de preferências individual e seleciona o candidato como o máximo de valores de votação. A regra de pluralidade atribui o valor 1 ao candidato mais preferido de cada votante e 0 a todos os outros. A regra de Borda atribui valores de m (total de candidatos) até 1 de acordo com as preferências de cada votante. O conceito de vencedor de Condorcet, o candidato preferido dos votantes de acordo com a maioria estrita dos votantes. Existem perfis que não apresentam um vencedor de Condorcet. É óbvio que quando existe um vencedor de Condorcet este é único. Uma regra consistente com Condorcet é uma votação que elege um vencedor de Condorcet quando este existe.

Atribuição de Recursos e Divisão Justa

A atribuição de recursos dos bens indivisíveis pretende assignar itens do um conjunto finito R aos membros de um conjunto de agentes N dadas as suas preferências sobre todos os possíveis conjuntos de bens. Na atribuição centralizada a distribuição é determinada por uma autoridade central à qual, previamente, os agentes fizeram saber das suas preferências. Numa distribuição descentralizada os agentes negociam, comunicam os seus interesses e trocam ou fazem o intercâmbio de bens em várias rondas, possivelmente de uma maneira multilateral. Podem ser distinguidos para avaliar a qualidade da distribuição de recursos nomeadamente a eficiência e equidade. O método fundamental para o critério de eficiência é a eficiência de Pareto: uma atribuição de recursos deve ser tal que não exista uma atribuição alternativa que seja melhor para uns agentes sem ser pior para qualquer um dos outros. Um exemplo de equidade é a *envy-freeness*: uma atribuição é *envy-free* se e só se nenhum agente pretende os bens de um outro.

Formação de Coligações

Em várias ocasiões os agentes não competem, mas cooperam para realizar uma tarefa mais eficazmente. Supondo que um agente X é recompensado com 10 quando realiza uma tarefa

individualmente e um agente Y é recompensado com 20. Se ao colaborarem o ganho poderá ser 50. A área estuda, tipicamente, duas questões: como irá se formar a coligação para um problema e como o excesso será dividido entre os diversos membros. O tema central é a noção de estabilidade: um agente não deve ter incentivo para abandonar a coligação. Estas questões são estudadas no campo da teoria de jogos competitivos [79] e diferentes conceitos de solução foram introduzidos. Por exemplo, o mais forte deste, conhecido como *núcleo* requer que nenhuma outra coligação seja mais benéfica para os seus membros.

Agregação de Julgamentos e Junção de Crenças

O campo de agregação de julgamentos pretende estudar como um grupo de indivíduos devem agregar os julgamentos individuais de cada membro em algumas proposições interligadas em julgamentos coletivos sobre estas. Estes problemas podem ocorrer em diferentes campos de decisões em grupos. A junção de crenças é um problema relacionado em que investiga a forma de agregar um número de bases de crenças numa base coletiva.

Sistemas de Escalonamento

Este sistema é a variação da escolha social clássica onde um conjunto de agentes e o conjunto de alternativas coincidem. A mais conhecida família de tais sistemas é o sistema de escalonamento de páginas no contexto dos motores de busca online.

9.3. CONCLUSÃO

Neste capítulo abordamos, resumidamente, o conceito de decisão através da participação de vários elementos de um grupo. Este partiu, como muitos dos conceitos abordados ao longo deste trabalho, da observação do comportamento natural. Abordamos algumas das dificuldades encontradas nos diversos sistemas de decisão bem como a sua transição para um sistema computacional.

10. BASES PARA CONSTRUIR MODELOS DE CONHECIMENTO COLETIVO NO PARADIGMA SIMBÓLICO

Algumas bases para permitir a construção de um modelo que suporte a representação do conhecimento coletivo foram apresentadas e nos capítulos anteriores. Este capítulo pretende sintetizar e apresentar os conceitos. A ideia do trabalho aparece após considerar a possibilidade de o conhecimento num grupo ser mais do que apenas a soma do conhecimento de cada elemento no grupo. Esta é baseada nas observações relatadas na Parte II do trabalho. A observação da natureza apresenta-nos várias proposições que sustentam a ideia de existir um conhecimento coletivo utilizado por um grupo para atingir os seus fins. Vários algoritmos desenvolveram-se baseando-se no comportamento existente no mundo natural. Na natureza vários animais funcionam em conjunto promovendo ações e tarefas que vão mais além das capacidades de apenas um elemento do grupo. O que indica uma capacidade de organização e distribuição de tarefas que parece improvável à primeira vista. No caso estudado das formigas/térmitas a informação é passada através da libertação das feromonas no caminho percorrido. Os elementos que passam são compelidos a seguir o percurso em que esta marcação é mais forte reforçando, por sua vez, a marcação do caminho. No caso das abelhas estas argumentam o seu caso através do processo de comunicação através da sua dança. As abelhas que são convencidas pela dança e qualidade da amostra de pólen seguem para a fonte de alimentação indicada pela abelha inicial e por sua vez irão realizar a dança tentando convencer as outras da qualidade da sua opção. Com a cuidadosa observação e estudo várias técnicas foram desenvolvidas para a resolução de diversos problemas. Apresentamos vários exemplos deste tipo de algoritmos que fazem uso do que se pode considerar conhecimento emergente para a resolução de várias questões computacionais. Nos capítulos referentes discutimos alguns exemplos de técnicas baseadas na natureza e incluímos um estudo sobre como uma organização artificial criada pelos seres humanos pode, também, ter um comportamento semelhante. Na Parte I introduzimos a Inteligência Artificial e o conceito de Agente. Estes serviram para suportar a ideia de elementos que realizam ações e fazem escolhas. A Inteligência Artificial desde o seu início ocupou-se de tentar estabelecer como os sistemas computacionais podem realizar as suas funções como de seres humanos se tratassem. O objetivo é conseguir sistemas mais capazes de realizar as suas funções. O conceito de Agente permite tratar dos problemas através da agregação de vários elementos computacionais em vez da centralização num só elemento. Depois abordamos algumas das ferramentas possíveis para a representação e raciocínio. Para a representação utilizamos a FOL. Esta é um sistema lógico que permite representar conhecimento e raciocinar sobre os mesmos de forma rigorosa e não ambígua. Para responder ao facto de existirem vários agentes a interagirem que podem ter diferentes visões sobre o ambiente recorremos à Teoria da Decisão que proporciona uma plataforma para lidar com a escolha em grupo

e como chegar a uma decisão em grupo. Como em todos os grupos a comunicação entre os elementos é importante, os SMAs têm vindo a tentar responder a estas questões da comunicação e cooperação entre os agentes. E dentro desta comunicação temos o exemplo das abelhas de tentar levar as outras à fonte de alimento. Como se pode concluir existe um processo em que um elemento argumenta da sua justiça para que os outros elementos do grupo sigam a sua visão. Este é um paralelo com a argumentação humana em que alguém tenta convencer o público a ver da sua opinião. Assim podemos abordar a Argumentação como ferramenta utilizada pelos agentes na sua interação com outros agentes. A alteração das crenças de um agente é suportada pela Revisão de Crenças.

10.1. AGENTES

O modelo de agente para o modelo é baseado na arquitetura BDI apresentada por Wooldridge, capítulo 8.5, com a adição da componente relativa ao conhecimento. Em [78] é discutida arquitetura desse tipo de agente. Em [87] o autor discute a abrangência e necessidade de um agente deste tipo para as necessidades atuais. No caso de um sistema multiagente, discutido no capítulo 3, a comunicação e cooperação apresenta-se com várias soluções. Assim o agente será capaz de interagir com o ambiente e com os outros agentes para construir o seu conhecimento. A interação dos agentes irá fazer o conhecimento do grupo aumentar. Com a capacidade de aprendizagem estes irão refinar o seu conhecimento e otimizar as suas ações. Este comportamento não é inédito como podemos constatar no Capítulo 3. Da observação do mundo natural verifica-se que os animais, em colónias, trabalham em conjunto para um objetivo comum e, que, mesmo com recursos limitados o conseguem fazer. Isto demonstra que a ideia do conhecimento coletivo não é apenas uma ideia, mas realidade. Assim o campo dos SMAs permite suportar a parte de funcionamento em grupo de vários agentes com recursos limitados. Os agentes BDI, com as alterações discutidas em [78] interpretam este cenário de forma adequada. De [87] podemos verificar uma mudança no sentido de incluir os resultados da aprendizagem com representação do conhecimento com uma maior abrangência com inteligência artificial.

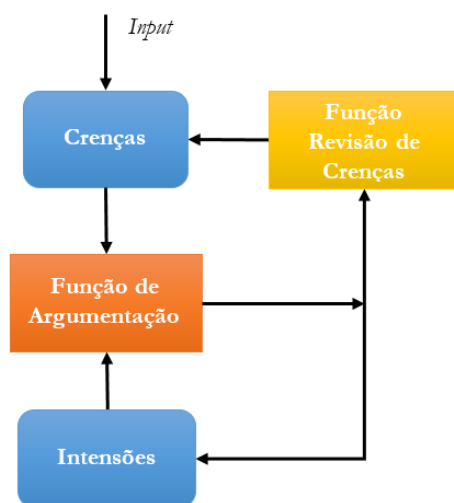


Figura 10.1 – Modelo de agente simplificado

A Figura 10.1 apresenta, de forma simplificada, o modelo do agente proposto. Na situação apresentada o que o agente absorve do exterior (quer seja pela observação do ambiente quer pela interação com outros agentes) são consideradas. O conjunto de Intensões contém os estados motivacionais do agente. A função de Argumentação irá selecionar a intensão do agente. Esta irá ser executada se tiver cabimento dentro da validade do conhecimento. A função de Argumentação será responsável pela verificação das intensões do agente perante as informações recebidas do ambiente. A função de Revisão de Crenças entra em ação quando as Crenças entram em contradição. Dessa forma o modelo permite o agente atualizar as suas crenças bem como argumentar da sua razão.

10.2. REPRESENTAÇÃO E RACIOCÍNIO

O conhecimento é uma informação armazenada ou modelos utilizados por uma pessoa ou máquina para interpretar, prever e responder apropriadamente ao mundo exterior. Com tal temos de ponderar duas características: a informação explícita e como a informação é codificada para posterior utilização. A representação de conhecimento tem um objetivo: fornecer uma adequada representação do conhecimento. Esta representação irá afetar quão bem a solução será adequada ao problema. O paradigma simbólico apresenta uma boa solução pois o conhecimento é apresentado através de frases declarativa e as consequências são deduzidas através de métodos de raciocínio lógico.

Assim temos como exemplo o seguinte as seguintes informações:

- $\forall x \forall y \text{irmão}(x, y) \Rightarrow \text{parente}(x, y)$
- $\forall x \forall y \forall z \text{pai}(z, x) \wedge \text{pai}(z, y) \Rightarrow \text{irmão}(x, y)$

Agora vemos a aplicação:

- $\text{pai}(\text{João}, \text{Maria})$
- $\text{pai}(\text{João}, \text{Eduardo})$

Pelo que a Maria e o Eduardo são parentes. Esta conclusão advém dos três pontos que são necessários para uma abordagem simbólica:

- Identificar o conhecimento do domínio (modelo do problema);
- Representar o mesmo através de uma linguagem formal de representação;
- Implementar um mecanismo de inferência para utilização do conhecimento.

A FOL é uma ferramenta poderosa para a Representação e Raciocínio pela que a sua utilização no modelo é óbvia. Os exemplos apresentados servem para ilustrar como poderemos utilizar as suas propriedades para que o conhecimento seja fiável e que o raciocínio seja fundamentado.

10.3. DINÂMICA DO CONHECIMENTO

O que se pretende é o conhecimento coletivo, mas falamos apenas da representação do conhecimento de um agente. Para isso os diversos agentes terão de comunicar e trocar impressões de acordo com o seu conhecimento. Para tal o processo de argumentação, já descrito, anteriormente servirá como base para os agentes, entre si, possam trocar impressões das visões individuais. Quando isto acontece

o agente terá de reavaliar o seu conhecimento. Aqui entre em jogo e revisão de crenças que dispõe de mecanismos de suporte à mudança de crenças internas de um agente.

Como suporte do Conhecimento Coletivo abordamos a Teoria da Decisão e esta deu-nos as bases para um conhecimento coletivo. Se os agentes têm um modelo do mundo, o seu conhecimento, seria possível ter uma agente que questionasse todos os agentes do grupo sobre uma determinada característica. Esse agente teria, então, a resposta do grupo, não só de um agente, através das técnicas da Teoria da Decisão.

10.4. MODELO

Com todas as partes do modelo identificadas a Figura 10.2 ilustra uma possível efetivação do modelo.

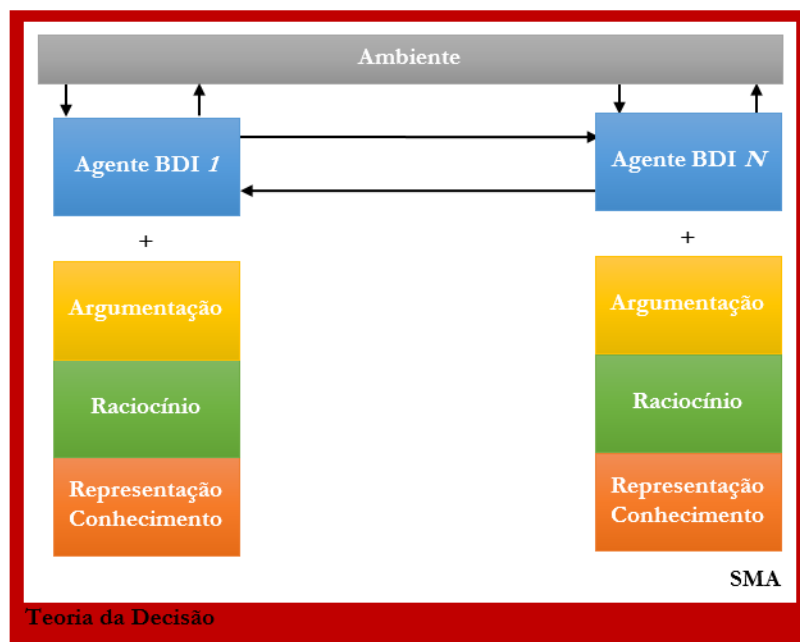


Figura 10.2 – Modelo base

O modelo apresenta um conjunto de agentes com a arquitetura descrita por Wooldridge [78]. Estes serão equipados com argumentação, raciocínio e representação do conhecimento. Assim serão capazes de atualizar o seu conhecimento através da interação com o ambiente envolvente. Esta atualização procede através do processo de Revisão de Crenças quando encontra informação que provoca contradição no seu conhecimento. Como consideramos um sistema multiagente os agentes interagem trocando informação do seu conhecimento. O processo de Argumentação pode ocorrer de forma estabelecer qual a informação que deve ser mantida. Para manter a informação utilizamos a FOL como linguagem o paradigma simbólico como sistema de base. O que permite o agente raciocinar sobre o seu conhecimento. A Teoria da Decisão permite que os agentes possam tomar decisões em conjunto e suportar o conhecimento coletivo presente no sistema.

10.5. CONCLUSÃO

Neste capítulo resumimos a base para a construção de um modelo de conhecimento coletivo. Esta base apresenta como linguagem a FOL que permite uma correta descrição tanto dos dados como do processo. A escolha do paradigma simbólico deve-se a que este tem a capacidade de expressão que falta a outros paradigmas. A transição do domínio do problema para os símbolos é feita deste o tempo da arte rupestre. A capacidade do ser humano de abstrair os conceitos permite uma manipulação da informação mais simples. Em termos computacionais não é assim tão direta pois é necessário que o seu domínio esteja corretamente representado para que seja possível tomar decisões não ambíguas.

Ao longo deste trabalho seguimos da observação dos elementos naturais para criarmos uma visão que suporte um modelo. Esse surgiu da observação que na relação de um conjunto de elementos pode surgir conhecimento. Como exemplo mostramos alguns elementos do mundo natural que deram origem a várias aplicações que resolveram alguns problemas existentes. Mostramos também o trabalho apresenta um conceito semelhante numa empresa em que se cria um conhecimento através da relação dos seus elementos. Assim investigamos bases para um sistema que capturasse estas nuances e fornecesse a possibilidade de inquirir este sistema para retirar informações relevantes. A literatura apresenta diversos paradigmas da inteligência artificial para retirar conclusões sobre como melhor representar a situação. O paradigma simbólico mostra-se capaz de suportar esta ideia apesar de apresentar algumas complexidades comparados a outros paradigmas.

O paradigma simbólico tem a grande vantagem que através da inferência lógica é possível definir formalmente o processo o que garantem a robustez das conclusões. Este sistema formal, como todos, apresenta a desvantagem de ser restritiva: é necessário ter todas as condições para se realizar a inferência. Sendo incapazes de lidar com informação que não seja parcial, incompleta ou aproximada, mas existem técnicas capazes de relevar estes problemas [87]. Como resposta utilizamos os agentes que possuem informação completa para a sua realidade. Através da interação entre os agentes esta informação irá ser complementada para que o conhecimento coletivo seja o mais correto.

Este é um objetivo complexo pois os agentes estão presentes num ambiente em que necessitam de o conhecer e tomar ações com base no seu conhecimento. Além de suportar a representação de conhecimento é necessário que seja capaz de raciocinar e argumentar com os restantes elementos do grupo.

Devemos referir que muito, ainda, poderá ser explorado devido à constante evolução e descoberta de novas técnicas. Mas, consideramos o trabalho como relevante pois apresenta uma perspetiva do grupo em vez da perspetiva do agente.

11. CONCLUSÕES E TRABALHO FUTURO

Neste trabalho mostramos a viabilidade da construção de um modelo de conhecimento coletivo no paradigma simbólico. Para tal utilizamos a ideia introduzida por vários algoritmos de inspiração biológica como o ACO, PSO e Bee Algorithm. O modelo SECI também mostra tal capacidade numa construção genuinamente humana como uma empresa. Para conseguir tal objetivo abordamos paradigmas da inteligência artificial com o intuito de escolher o mais indicado. Optamos pelo paradigma simbólico pois é que permite a representação formal do conhecimento e raciocínio. Sendo o paradigma que consegue definir o porquê das suas conclusões. Abordamos várias técnicas/algoritmos para a representação e raciocínio para base do modelo. Através da pesquisa percorremos um caminho que nos levou a várias áreas. Tal computação inspirada na biologia, sistemas multiagentes, lógica, revisão de crenças, argumentação, representação de conhecimento, raciocínio o que nos permitiu ter uma visão abrangente sobre os campos que inteligência artificial pode beneficiar e/ou retirar novas ideias. Sendo que uma área de estudo que cubra todos estes conceitos parece ser rara.

Resumimos a contribuição do trabalho nos seguintes pontos: uma visão sobre um conjunto de áreas aparentemente não relacionadas, mas que contribuem para diversas situações; trazer o paradigma simbólico a discussão atual; apresentação de uma nova visão sobre o tema e a junção de vários campos numa solução.

Como trabalho futuro apontar a construção da plataforma de suporte para a construção dos modelos, sumariada nos seguintes pontos:

- Expansão dos agentes argumentativos para terem a capacidade de armazenar o conhecimento;
- Criar a arquitetura da plataforma;
- Desenvolver a linguagem formal para a plataforma;
- Desenvolver modelos mistos.

As bases apresentadas no trabalho permitem sustentar o modelo. A construção da plataforma, e linguagem, permitirá interligar os elementos apresentados. Estes terão, obviamente, de ser aprofundados e adequados tendo em conta o fim proposto e características próprias.

12. BIBLIOGRAFIA E REFERÊNCIAS

-
- [01] Gardner, Howard, Basic Books, *Frames of Mind: The Theory of Multiple Intelligences*, Third Edition, 2011. First Edition: 1983
-
- [02] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, New York, 3rd edition, 2010. 1st edition: 1995.
-
- [03] A. Carpi, C.A. Brebbia, *Design & Nature V: Comparing Design in Nature with Science and Engineering*, 2010, pp 275-277
-
- [04] A. Bonner, *The Art and Logic of Ramon Llull: A User's Guide*, 2007, pp 290
-
- [05] Asimov, Isaac (September 1979). "The Vocabulary of Science Fiction.". *Asimov's Science Fiction*.
-
- [06] McCarthy *et al.* "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", 1955
-
- [07] McCorduck, Pamela (2004), *Machines Who Think* (2nd ed.), Natick, MA: A. K. Peters, Ltd., ISBN 1-56881-205-1, pp. 161–170.
-
- [08] John McCarthy. "Recursive Functions of Symbolic Expressions and Their Computation by Machine, Part I"
-
- [09] Yu, V.L. et al. (1979). "Antimicrobial selection by a computer: a blinded evaluation by infectious disease experts", *Journal of the American Medical Association* 242 (12): 1279–1282.
-
- [10] *The AI Business: The commercial uses of artificial intelligence*, ed. Patrick Winston and Karen A. Prendergast. ISBN 0-262-73077-4
-
- [11] *A Learning Evolutionary Trading System LETS*, Sonia Schulenburg et al, 2007
-
- [12] Moore, J.H. & Holmes, J.H. *BioData Mining* (2016) 9: 15. doi:10.1186/s13040-016-0092-6
-
- [13] Xavier Llorà et al, *Towards Better than Human Capability in Diagnosing Prostate Cancer Using Infrared Spectroscopic Imaging*, 2007
-
- [14] Wooldridge M. *An Introduction to MultiAgent Systems*, John Wiley & Sons, London, 2nd edition, 2009. 1st edition 2001.
-
- [15] Maes, P., *The agent network architecture (ANA)*. *SIGART Bulletin*, 2(4):pp115-120, 1991
-
- [16] <http://www.agentlink.org> [Online].
-
- [17] L. P. Reis, Phd Thesis, *Coordenação em Sistemas Multi-Agente: Aplicações na Gestão Universitária e Futebol Robótico*, FEUP, 2003
-
- [18] Sage A. P., *Concise Encyclopedia of Information Processing in Systems & Organizations*, Pergamon Pr, 1990
-
- [19] Haykin, Simon, *Neural networks: a comprehensive Foundation*, 2nd Edition, Prentice Hall, Inc. 1999
-
- [20] Fischler e Firschein, 1987. "Intelligence : the eye, the brain, and the computer", Addison-Wesley
-
- Rumelhart, D. E., & McClelland, J. L. (1986). *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*. Cambridge, MA: MIT Press.
-
- [21] McClelland, J. L., & Rumelhart, D. E. (1986). *Parallel Distributed Processing. Explorations in the Microstructure of Cognition. Volume 2: Psychological and Biological Models*. Cambridge, MA: MIT Press.
-
- [22] Simões A., Costa E., "Inteligência Artificial Fundamentos e Aplicações" FCA – Editora de Informática, Lda. Lisboa 2004
-

- [23] Hopfield, J. J. (1982). “Neural networks and physical systems with emergent collective computational abilities”. Proc. NatL Acad. Sci. USA Vol. 79, pp. 2554-2558, April 1982
Biophysics
- [24] “Alterações cromossômicas”, EuroGentest,
URL = < <http://www.eurogentest.org/index.php?id=461>>.
- [25] Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO
Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30
(1989)
- [26] D. Merkle, M. Middendorf, “Swarm Intelligence”, Search Methodologies, pp. 401-435,
2005
- [27] Floreano, Dario and Mattiussi Claudio, Bio-Inspired Artificial Intelligence – Theories,
Methods, and Technologies, The MIT Press, 2008
- [28] Biomolecular Computing Systems Harish Chandran, Sudhanshu Garg, Nikhil
Gopalkrishnan, John Reif Department of Computer Science, Duke University, Durham
NC 27708
- [29] Biocomputing based on particle disassembly, Maxim P. Nikitin, Victoria O. Shipunova,
Sergey M. Deyev and Petr I. Nikitin, Nature Nanotechnology 17/ 08/2014
- [30] Social insect [em linha: <http://www.britannica.com/animal/social-insect>]
- [31] Social Insects [em linha: <https://www.cals.ncsu.edu/course/ent425/tutorial/Social/>]
- [32] M. Dorigo, Optimization, Learning and Natural Algorithms, PhD thesis, Politecnico di
Milano, Italy, 1992.
- [33] Dorigo M., Stützle T., *Ant Colony Optimization*, A Bradford Book, The MIT Press, 2004
- [34] J.-L. Deneubourg et al. “The Self-organizing Exploratory Pattern of Argentine Ant” in
Journal of Insect Behavior, Vol. 3, 1990, pp. 159-168
- [35] M. Dorigo, V. Maniezzo, A. Colomi, “The Ant System: Optimization by a colony of
cooperating agents” in *IEEE Transactions on Systems, Man, and Cybernetics–Part B*,
Vol.26, No.1, 1996, pp.1-13
- [36] Pham DT, Ghanbarzadeh A, Koc E, Otri S, Rahim S and Zaidi M. The Bees Algorithm.
Technical Note, Manufacturing Engineering Centre, Cardiff University, UK, 2005.
- [37] Von Frisch K. Bees: Their Vision, Chemical Senses and Language. (Revised edn) Cornell
University Press, N.Y., Ithaca, 1976.
- [38] Camazine S, Deneubourg J, Franks NR, Sneyd J, Theraula G and Bonabeau E. Self-
Organization in Biological Systems. Princeton: Princeton University Press, 2003.
- [39] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. *Micro
Machine and Human Science*, 1995. MHS '95., Proceedings of the Sixth International
Symposium on, pages 39-43, Oct 1995
- [40] J. Kennedy and R. C.Eberhart, “Particle Swarm Optimization”, *IEEE International
Conference on Neural Networks*, 1995, pp.1942-1948
- [41] Ikujiro Nonaka, Hirotaka Takeuchi. *The Knowledge-creating Company: How Japanese
Companies Create the Dynamics of Innovation*, Oxford University Press, 1995
- [42] Hirotaka Takeuchi, *The New Dynamism of the Knowledge-Creating Company*, Japan
Moving Toward a More Advanced Knowledge Economy: Advanced Knowledge-
Creating Companies, by Hirotaka Takeuchi and Tsutomu Shibata. Washington, D.C.:
World Bank Institute (WBI), 2006.
- [43] *Handbook of Knowledge Representation*, Editors van Harmelen F., Lifschitz V., Porter
B., Foundations of Artificial Intelligence, Series Editors Hendler J., Kitano H., Nebel B.,
ELSIVIER 2008
- [44] Konar A., *Artificial Inteligence and Soft Computing – Behavioral and Cognitive
Modeling of the Human Brain*, CRC Press LLC, 2000
- [45] *Knowledge Representation and Reasoning*, Brachman R. J., Levesque H. J., Elsevier,
2004
- [46] Hector J. Levesque and Gerhard Lakemeyer. *The Logic of Knowledge Bases*. MIT Press,
Cambridge, 2000.
- [47] I. Levi, “Subjunctives, dispositions and changes”, *Synthese*, vol. 34, n°. 4, pp. 423-455,
1977.

- [48] I. Levi, *The Enterprise of Knowledge*, Cambridge:MIT Press, 1980.
- [49] W. L. Harper, “Ramsey test conditionals and iterated belief change”, in *Foundations and Philosophy of Epistemic Applications of Probability Theory*, Dordrecht, D. Reidel Publishing Co., 1976, pp. 117-136.
- [50] C. Alchourrón, P. Gärdenfors and D. Makinson, “On the logic of theory change: Partial meet contraction and revision functions”, *The Journal of Symbolic Logic*, vol. 50, pp. 510-530, 1985.
- [51] E. Fermé and S. Hansson, “AGM 25 Years”, *Journal of Philosophical Logic*, vol. 40 n.º 2, pp. 295-331, 2011.
- [52] S. O. Hansson, “Logic of Belief Revision”, in *The Stanford Encyclopedia of Philosophy* (Fall 2011 Edition), E. N. Z. (ed.), Ed.
- [53] E. L. Fermé, “Tópicos Avançados em Revisão de Crenças,” 2008.
- [54] C. Alchourrón and D. Makinson, “On the logic of theory change: Safe contraction,” *Studia Logica*, vol. 44, p.405–422, 1985.
- [55] S. O. Hansson, “A Survey of Non-Prioritized Belief Revision,” *Erkenntnis*, vol. 50, pp. 413-427, 1999.
- [56] S. O. Hansson, *A Textbook of Belief Dynamics. Theory Change and Database Updating*, Dordrecht: Kluwer Academic Publishers, 1999.
- [57] H. Sven Ove, *A Textbook of Belief Dynamics. Theory Change and Database Updating*, Dordrecht: Kluwer Academic Publishers, 1999.
- [58] C. Rapp, “Aristotle’s Rhetoric”, *The Stanford Encyclopedia of Philosophy* (Spring 2010 Edition), [em linha: <http://plato.stanford.edu/archives/spr2010/entries/aristotle-rhetoric>]
- [59] D. Walton, “Argumentation Theory: A Very Short Introduction “, *Argumentation in Artificial Intelligence*, springer US, 2009, pp 1-22.
- [60] J. L. Pollock, “Knowledge and Justification”, Princeton University Press, 1974.
- [61] D. Nute, “Defeasible Reasoning”, in *Aspects of Artificial Intelligence*, Noewell, Kluwer Academic Publishers, 1988, pp. 251-288.
- [62] C. I. Chesñevar and G. R. Simari, “Computational Models for Argumentation in Multiagent Systems,” in *EASSS*, Utrecht, Netherlands, 2005.
- [63] J. Doyle, “A Truth Maintenance System,” in *Artificial Intelligence*, 1979, pp. 12:231-272.
- [64] G. Simari and R. P. Loui, “A Mathematical Treatment of Defeasible Reasoning and its Implementation,” *Artif. Intell.*, pp. 125-157, 1992.
- [65] G. Vreeswijk, “Abstract argumentation systems,” *Artif. Intell.*, pp. 225-279, 1997
- [66] P. M. Dung, “On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games,” *Artif. Intell.*, pp. 321-358, 1995.
- [67] A. J. García and G. R. Simari, “Defeasible logic programming: an argumentative approach,” *Theory and Practice of Logic Programming*, vol. 4, pp. 95-138, 2004.
- [68] M. A. Falappa, G. Kern-Isberner and G. R. Simari, “Belief revision and argumentation,” in *Argumentation in artificial intelligence*, US: Springer, 2009, p. 341–360.
- [69] J. d. Kleer, "An assumption-based TMS," *Artificial Intelligence*, vol. 28, no. 2, pp. 127-162, 1986.
- [70] J. L. Pollock and A. S. Gillies, "Belief Revision and Epistemology," *Synthese*, vol. 122, no. 1, pp. 69-92, 2000.
- [71] M. A. Falappa, G. Kern-Isberner and G. R. Simari, "Belief Revision, Explanations and Defeasible Reasoning," *Artificial Intelligence*, vol. 141, pp. 1-28, 2002.
- [72] F. Paglieri and C. Castelfranchi, "The Toulmin Test: Framing Argumentation within Belief Revision Theories," in *Arguing on the Toulmin Model*, Springer Netherlands, 2006, pp. 359-377
- [73] N. D. Rotstein, M. O. Moguillansky, M. A. Falappa, A. J. García and G. R. Simari, "Argument Theory Change: Revision Upon Warrant," in *Proceedings of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, The Netherlands, 2008.

-
- [74] G. Boella, C. d. C. Perera, A. Tettamanzi and L. v. d. Torre, "Dung Argumentation and AGM Belief Revision," in *ArgMAS 2008*, Estoril, Portugal, 2008.
-
- [75] M. O. Moguillansky, N. D. Rotstein, M. A. Falappa, A. J. García and G. R. Simari, "Argument theory change applied to defeasible logic programming," in *Proceedings of the 23rd conference on Artificial intelligence, AAAI 2008*, 2008.
-
- [76] A. Fuhrmann and S. O. Hansson, "A Survey of Multiple Contractions," *Journal of Logic, Language, and Information*, vol. 3, pp. 39-75, 1994.
-
- [77] G. Boella, C. Pereira, A. Tettamanzi and L. van der Torre, "Making Others Believe What They Want," in *Artificial Intelligence in Theory and Practice II*, Springer Boston, 2008, pp. 215-224.
-
- [78] M. Falappa, A. García and G. Simari, "Belief dynamics and defeasible argumentation in rational agents," in *NMR*, 2004, pp. 164-170.
-
- [79] M. E. Bratman, *Intention, Plans, and Practical Reason*, Cambridge, Massachusetts: Harvard University Press (Reprinted by CLSI Publications), 1987.
-
- [80] I. O. d. Nunes, "Implementação do Modelo e da Arquitetura BDI," PUC-Rio Departamento de Informática, Rio de Janeiro, Brasil, 2007.
-
- [81] M. Wooldridge, *Reasoning about Rational Agents*, MIT Press, 2000.
-
- [82] A. S. Rao e M. P. Georgeff, "BDI Agents: From Theory to Practice," em *Proceedings of the First International Conference on Multiagent Systems*, San Francisco, 1995.
-
- [83] M. Wooldridge, *Intelligent agents*, 1999.
-
- [84] Resnik M. D., "Choices An Introduction to Decision Theory" Minnesota University Press, 2000
-
- [85] List, Christian, "Social Choice Theory", *The Stanford Encyclopedia of Philosophy* (Winter 2013 Edition), Edward N. Zalta (ed.), URL = <http://plato.stanford.edu/archives/win2013/entries/social-choice/>.
-
- [86] Chevaleyre Y., Endriss U., Lang J., Maudet N., *A Short Introduction to Computational Social Choice*
-
- [87] Shoham Y, "Why Knowledge Representation Matters", *Communications of the ACM*, Vol. 59 No. 1, Pages 47-49, 10.1145/2803170, 2016
-
- [B1] Gonçalves C., "Argumentative Systems in Robots", Master Dissertation, UMa, 2012
-

A Nossa Universidade

Colégio dos Jesuítas
Rua dos Ferreiros - 9000-082, Funchal

Tel: +351 291 209400
Fax: +351 291 209410
Email: gabinetedareitoria@uma.pt